

Trabajo de Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

**PROPUESTA DE MEJORA DE LA GESTIÓN DE
RESIDUOS APLICANDO TECNOLOGÍA RFID**

MEMORIA

Autor: Àlex Gómez Segura
Director: José Luis Eguía Gómez

Convocatoria: 04 2020



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resumen

El objeto del presente trabajo es el análisis del desarrollo de un prototipo de contenedor doméstico inteligente que recomienda la fracción donde depositar el residuo que el usuario quiera introducir mediante la tecnología de identificación por radiofrecuencia. Se examinan los proyectos previos y se explican las tecnologías vinculadas al dispositivo, como la tecnología de identificación por radiofrecuencia y el prototipado de circuitos electrónicos mediante Arduino. Mediante un análisis comparativo de las alternativas tecnológicas se determinan los componentes a utilizar para satisfacer los requerimientos del prototipo, atendiendo también a criterios económicos y de disponibilidad por parte de la Universidad. Seguidamente, se expone la estructura y funcionamiento del referido prototipo, valorando sus prestaciones mediante la realización de pruebas. Finalmente, se verifica el cumplimiento de los objetivos planteados y se proponen líneas futuras de desarrollo o mejora del contenedor.

Overview

The purpose of this work is the analysis of the development of a prototype of an intelligent domestic container that recommends the fraction where to deposit the waste that the user wants to introduce using radio frequency identification technology. Previous projects are reviewed and technologies related to the device, such as radio frequency identification technology and prototyping of electronic circuits using Arduino, are explained. Through a comparative analysis of the technological alternatives, the components to be used to satisfy the requirements of the prototype are determined, also taking into account economic and availability criteria by the University. Next, the structure and operation of the aforementioned prototype is exposed, evaluating its benefits by carrying out tests. Finally, compliance with the stated objectives is verified and future lines of development or improvement of the container are proposed.

Resum

L'objecte d'aquest treball és l'anàlisi del desenvolupament d'un prototip de contenidor domèstic intel·ligent que recomana la fracció on dipositar el residu que l'usuari vulgui introduir mitjançant la tecnologia d'identificació per radiofreqüència. S'examinen els projectes previs i s'expliquen les tecnologies vinculades al dispositiu, com la tecnologia d'identificació per radiofreqüència i el prototipat de circuits electrònics mitjançant Arduino. Mitjançant una anàlisi comparativa de les alternatives tecnològiques es determinen els components a utilitzar per satisfer els requeriments del prototip, atenent també a criteris econòmics i de disponibilitat per part de la Universitat. Seguidament, s'exposa l'estructura i funcionament de l'esmentat prototip, valorant-ne les prestacions mitjançant la realització de proves. Finalment, es verifica el compliment dels objectius plantejats i es proposen línies futures de desenvolupament o millora del contenidor.

Sumario

1. GLOSARIO	9
2. PREFACIO	12
2.1. Origen del proyecto	12
2.2. Motivación	13
2.3. Requerimientos previos.....	13
3. INTRODUCCIÓN	14
3.1. Definición del problema y justificación del Trabajo.....	15
3.2. Objetivos del proyecto	17
3.2.1. Objetivo principal.....	17
3.2.2. Objetivos específicos	17
3.3. Alcance y estructura del proyecto	18
4. ESTADO DEL ARTE	19
4.1. Proyectos previos.....	19
4.1.1. <i>A Smart Waste Management with Self-Describing objects</i>	19
4.1.2. Dispositivos inteligentes: <i>Smart Trash Bin</i> y RFM.....	20
4.1.3. Etiquetas RFID en los productos alimentarios	21
4.1.4. PREMETS 2025.....	22
4.2. Identificación automática	23
4.3. Internet de las cosas	24
4.4. Identificación por radiofrecuencia.....	24
4.4.1. Componentes de un sistema RFID	25
4.4.1.1. El tag.....	27
4.4.1.2. El lector	27
4.4.1.3. <i>Middleware</i> y base de datos.....	28
4.4.1.4. Programador RFID	28
4.4.2. Comparativa entre la tecnología RFID y el código de barras	28
4.4.3. Estándares y legislación	30
4.4.3.1. GS1 - EPCglobal.....	30
4.4.3.2. ISO	32
4.4.3.3. Legislación	33
5. DESARROLLO DE LA IDEA	34

5.1.	Definición del usuario.....	34
5.2.	Definición de requisitos y funcionalidades	35
5.3.	Alternativas de diseño.....	36
5.4.	Alternativas tecnológicas	37
5.4.1.	Placa de desarrollo	37
5.4.2.	Sistema RFID	41
5.4.2.1.	LF.....	42
5.4.2.2.	HF	44
5.4.2.3.	UHF.....	45
5.4.3.	Cantidad de residuos generados y nivel de llenado.....	46
5.4.4.	Conectividad a Internet	48
5.4.5.	Visualización de la información	50
6.	IMPLEMENTACIÓN DEL <i>HARDWARE</i>	52
6.1.	Alternativas tecnológicas escogidas	52
6.1.1.	Placa de desarrollo: Arduino UNO	52
6.1.2.	Sistema RFID	54
6.1.2.1.	LF.....	54
6.1.2.2.	HF	56
6.1.2.3.	UHF.....	58
6.1.3.	Sensor de ultrasonido: HC-SR05.....	60
6.1.4.	Módulo de conectividad a Internet: ESP-01	61
6.1.5.	Pantalla LCD.....	62
6.2.	Arquitectura final del prototipo	63
6.2.1.	Sistema RFID de LF	63
6.2.2.	Sistema RFID de HF.....	65
7.	IMPLEMENTACIÓN DEL <i>SOFTWARE</i>	67
7.1.	Funciones generales.....	67
7.1.1.	Función <i>ultrasonicSensor()</i>	67
7.1.2.	Función <i>connectWifi()</i>	68
7.1.3.	Función <i>printLCD()</i>	68
7.2.	<i>Software</i> sistema RFID de LF.....	69
7.3.	<i>Software</i> sistema RFID de HF	70
7.3.1.	<i>Software</i> de grabación de etiquetas.....	70
7.3.2.	<i>Software</i> de lectura de etiquetas.....	71
7.4.	Pruebas y simulaciones	72
7.4.1.	Prueba de rango de lectura.....	72

7.4.2.	Prueba de tiempo de lectura	73
7.4.3.	Prueba de detección del nivel de llenado.....	74
8.	VALORACIÓN ECONÓMICA	76
8.1.	Presupuesto del prototipo	76
8.1.1.	Set general.....	76
8.1.2.	Contenedor con sistema RFID de LF	76
8.1.3.	Contenedor con sistema RFID de HF	77
8.2.	Presupuesto del proyecto.....	77
9.	IMPACTO MEDIOAMBIENTAL	78
10.	PLANIFICACIÓN DEL TRABAJO	79
11.	CONCLUSIONES Y LÍNEAS FUTURAS	81
	AGRADECIMIENTOS	86
	BIBLIOGRAFÍA	87
	ANEXOS	89
	Anexo I: Encuesta definición del usuario	90
	Anexo II: Código <i>LecturaLF</i>	91
	Anexo III: Código <i>EscrituraHF</i>	95
	Anexo IV: Código <i>LecturaHF</i>	98
	Anexo V: Diagrama de Gantt.....	102

1. Glosario

AIDC: *Automatic Identification and Data Capture*. Tecnología para identificar un objeto automáticamente y extraer datos sobre el mismo.

ARDUINO: Plataforma de prototipado electrónico de código abierto.

ASCII: *American Standard Code for Information Interchange*. Sistema de codificación de caracteres alfanuméricos que asigna un número del 0 al 127 a cada letra, número o carácter especial recogidos. El ASCII extendido permite hasta 256 caracteres distintos

CÓDIGO EPC: *Electronic Product Code*. Código numérico estandarizado que se asigna a un objeto para identificarlo de forma unívoca.

COMUNIDAD MAKER: Comunidad de personas dedicadas a desarrollar proyectos tecnológicos.

EEPROM: *Electrically Erasable Programmable Read-Only Memory*. Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente

EPCglobal: empresa conjunta entre GS1 y GS1 US. Es una organización creada para lograr la adopción y estandarización en todo el mundo de la tecnología de Código de producto electrónico (EPC).

ETHERNET: Estándar de redes de área local para computadores con acceso al medio por detección de portadora, y con detección de colisiones.

ETSI: *European Telecommunications Standards Institute*. Organización europea independiente de estandarización.

FTDI: Dispositivo semiconductor para la conversión de transmisiones serie RS-232 o TTL a señales USB.

GPIO: *General Purpose Input/Output*. es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar por el usuario en tiempo de ejecución.

I2C: *Inter-Integrated Circuit*. Bus serie de datos que utiliza solo 2 cables, uno para el reloj y otro para el dato. Esto significa que el maestro y el esclavo envían datos por el mismo cable, el cual es controlado por el maestro, que crea la señal de reloj.

IDE: *Integrated Development Environment*. Entorno de programación consiste en un editor de código y un compilador.

IoT: *Internet of Things*. Interconexión digital de objetos cotidianos con Internet.

ISO: *International Organization for Standardization*. Organización internacional encargada de desarrollar estándares y normas.

LCD: *Liquid-Crystal Display*. Pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.

LED: *Light-Emitting Diode*. Diodo semiconductor que emite luz cuando se le aplica tensión.

LIDAR: *Light Detection and Ranging o Laser Imaging Detection and Ranging*. Dispositivo que permite determinar la distancia desde un emisor láser a un objeto o superficie utilizando un haz láser pulsado.

NFC: *Near-Field Communication*. Tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

OLED: *Organic Light-Emitting Diode*. Diodo que se basa en una capa electroluminiscente formada por una película de componentes orgánicos que reaccionan a una determinada estimulación eléctrica, generando y emitiendo luz por sí mismos.

ONS: *Object Name Service*. Base de datos global que permite la búsqueda de información de un producto o servicios relacionados a un EPC.

PLUGIN: Aplicación (o programa informático) que se relaciona con otra para agregarle una función nueva y generalmente muy específica.

PREMET25: Plan de gestión de residuos aprobado por el Área Metropolitana de Barcelona en 2019.

PROTOBOARD: Tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables el prototipado de circuitos electrónicos.

QR: *Quick Response code*. Módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional.

RAEE: Residuo de aparatos eléctricos y electrónicos.

RASPBERRY PI: ordenador de placa reducida, ordenador de placa única u ordenador de placa simple de bajo costo con el objetivo de estimular la enseñanza de informática en las escuelas.

RFID: *Radio Frequency IDentification*. Sistema de identificación, almacenamiento y recuperación de datos remoto mediante radiofrecuencia.

RSU: Residuos Sólidos Urbanos.

SHIELD: placa de circuito impreso que se coloca sobre la placa Arduino y se conecta a ella mediante el acoplamiento de sus pines sin necesidad de alguna otra conexión externa. Su función es de actuar como una placa complementaria, ampliando las capacidades de la placa Arduino Base.

SOC: *System On a Chip*. Circuito electrónico que integra todos los componentes necesarios en un ordenador y otros sistemas electrónicos.

SPI: *Serial Peripheral Interface*. Estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

TELETAC: Sistema de telepeaje interoperable implantado en todas las autopistas de peaje españolas.

UART: *Universal Asynchronous Receiver-Transmitter*. Dispositivo que controla los puertos y dispositivos serie.

USB: *Universal Serial Bus*. Bus de comunicaciones que sigue un estándar que define los cables, conectores y protocolos para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

VPN: *Virtual Private Network*. Tecnología de red que permite que el ordenador en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada.

WEREABLE: Objeto de uso diario que llevamos siempre encima al que se le ha incorporado un microprocesador.

WIFI: *Wireless Fidelity*. Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

2. Prefacio

En este capítulo se explica el origen del proyecto, los motivos de la elección de su objeto y los requerimientos previos necesarios para poder realizarlo.

2.1. Origen del proyecto

En agosto de 2018 inicié mi plan de movilidad Erasmus en Gotemburgo, Suecia. Durante mi estancia allí me desplazé utilizando las líneas de buses y tranvías de transporte público que la ciudad ofrecía. El acceso, tanto a buses como tranvías, se podía realizar mediante el uso de una tarjeta de transporte recargable o con el móvil gracias a la tecnología *contactless*.

Fui consciente de que desconocía la tecnología en la que se basan estos dispositivos para comunicarse inalámbricamente y sin necesidad de contacto. Además, mi curiosidad aumentó debido a que esta tecnología está presente muy habitualmente en nuestro día a día.

Más adelante descubrí que la tecnología que lo hace posible se denomina identificación por radiofrecuencia. Me pareció una tecnología muy interesante, útil y flexible, que podría ser empleada en muchos otros aspectos de nuestra vida cotidiana. Por ello me planteé realizar un proyecto basado en la identificación por radiofrecuencia para el Trabajo de Final de Grado.

Propuse dicho proyecto al profesor de mi universidad José Luis Eguía, actual director del presente Trabajo de Fin de Grado. Él confirmó que el proyecto era interesante y que debía darle un valor añadido a la tecnología. La contribución podía ser tanto una mejora tecnológica como añadir un segundo enfoque al uso de la identificación por radiofrecuencia. Me animó primero a realizar un estudio del arte de los proyectos relacionados con el uso de la identificación por radiofrecuencia. El estudio me permitiría obtener un punto de vista más amplio sobre las características y posibilidades de la tecnología. Surgió entonces la idea de diseñar un dispositivo para mejorar la gestión de residuos.

2.2. Motivación

Mi motivación principal para este proyecto es intentar contribuir positivamente en un problema social como es la gestión de los residuos, aprovechando los conocimientos adquiridos en el transcurso de mi grado en ingeniería industrial. Tales conocimientos entiendo que han de permitir generar aportaciones de utilidad a la sociedad en múltiples campos, entre los que se incluyen el de la gestión de residuos, que tiene importantes implicaciones sociales y medioambientales.

En efecto, teniendo en cuenta el momento en el que vivimos, en el que la lucha contra el cambio climático y la concienciación sobre la sostenibilidad de nuestro planeta cobran tanta importancia, realizar un proyecto que intente mejorar el medio ambiente es muy gratificante.

Una segunda motivación para desarrollar este proyecto es la ampliación de mis conocimientos en programación y en la tecnología de identificación automática por radiofrecuencia. Asimismo, el proyecto me permitirá discernir si me interesa este ámbito de la tecnología tanto como para, en un futuro, trabajar en ello y me iniciará en la metodología a seguir y los requisitos para llevar a cabo un proyecto.

Finalmente, una motivación secundaria que me impulsa es darle una visión comercial al proyecto y determinar si la propuesta que se planteará en el mismo o, al menos una similar, puede llegar a ser rentable.

2.3. Requerimientos previos

Para la realización de este proyecto es útil tener conocimientos básicos del lenguaje informático en el que se vaya a programar.

3. Introducción

Los desechos y residuos sólidos llamados comúnmente basura, son productos generados por la actividad humana, considerados como inútiles, indeseables o desechables. Son producto de las actividades humanas, al cual se le considera de valor igual a cero por quien lo desecha (Curcio, Blanco, Gil, & Rosa, 2015).

La gestión de los desechos tiene origen en la Antigua Grecia y desde entonces ha supuesto un reto para la humanidad. Sin embargo, en estos últimos años el problema se ha agravado. A causa del estilo de vida actual de la sociedad, en estos últimos años se ha llevado la generación de residuos a niveles extremos.

El avance de la tecnología en el último siglo ha generado un costo ambiental muy alto para la sociedad y la naturaleza, porque esta última está llegando al límite de su capacidad de abastecimiento regular de recursos renovables y de absorber los desechos resultantes del consumo de la sociedad (Cruz & Ojeda, 2013).

Las causas principales de dicho aumento son las siguientes:

- **Rotura del ciclo natural de circulación de la materia:** debido al desarrollo industrial se ha facilitado el consumo de productos manufacturados, compuestos por materiales más resistentes a los procesos de degradación natural (Albacete, 2006), como metales, plásticos, sustancias químicas, etc. Esto provoca un aumento de la acumulación de desechos y de su peligrosidad. *“El hombre actúa en la naturaleza como productor y consumidor, pero no como descomponedor”* (Albacete, 2006).
- **Era del consumo:** el consumo es el eje principal de nuestro sistema económico y cultural. El consumismo propicia el uso excesivo de embalajes y envases, y el consumo generalizado de productos de usar y tirar, lo que deriva en un aumento de la generación de desechos por persona.
- **Crecimiento demográfico:** la población mundial se ha triplicado desde los años 50, pasando de 2.530 millones de habitantes a, actualmente, 7.760 millones («Evolución de la población mundial hasta 2030 | Statista», s. f.). El rápido crecimiento de la población supone un constante aumento de residuos producidos.

Es fácil reconocer que la generación de desechos es, a nivel mundial, uno de los problemas ambientales más graves que los gobiernos deben enfrentar. Aunque en los últimos años ya se han tomado medidas, éstas han sido insuficientes y poco eficaces. La incorrecta gestión de los residuos ha agravado el agotamiento de los recursos naturales y ha incidido en la disminución de la calidad ambiental, repercutiendo en un impacto negativo sobre la salud de

los seres vivos. En la disminución de la calidad ambiental se incluyen grandes problemáticas como el cambio climático, la deforestación y la contaminación del aire, agua y suelo (Albacete, 2006).

Por ello es de vital importancia impulsar políticas medioambientales. Las actividades fundamentales de cualquier sistema de gestión de residuos en cualquier parte del mundo son, en orden de prioridad: la prevención, el reciclaje, la valorización energética y la deposición controlada de los residuos (ver Figura 1). La prevención o reducción de desechos incluye las políticas de concienciación de consumo responsable y práctica de la reutilización, el desarrollo circular sostenible de productos y el fomento de la educación y cultura ambiental. El reciclaje se define como el proceso de reintegración de los residuos en las cadenas económicas, convirtiéndolos de nuevo en recursos. La valorización energética consiste en extraer el valor energético de los residuos. Finalmente, si el desecho no ha podido ser reciclado o valorizado, ya sea por su composición, peligrosidad u origen, se deposita en un vertedero en condiciones de seguridad ambiental para evitar que sean una fuente de contaminación.

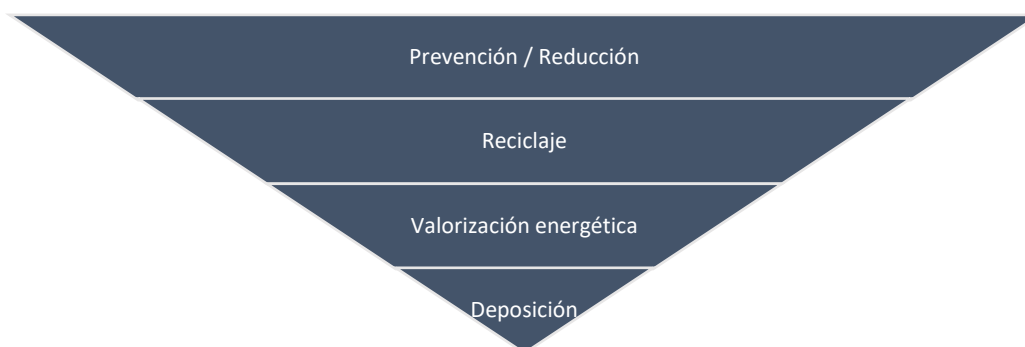


Figura 1. Fases del proceso de reciclaje

3.1. Definición del problema y justificación del Trabajo

El reciclaje es una de las prácticas de gestión de residuos más importantes y primordiales, acarreando cuantiosos beneficios.

Reciclar conlleva ahorrar materias primas, energía, agua y reducir las emisiones de gases de efecto invernadero. Además, supone una generación de empleo y tejido industrial («Beneficios del reciclaje de envases | Ecoembes», s. f.).

Hace ya muchos años que la práctica del reciclaje se realiza en gran parte del mundo, llevándose a cabo mediante sistemas de gestión de residuos implementados, habitualmente, por el gobierno de cada país. Desgraciadamente, solo algunos de los desechos son reciclables. Además, el proceso de reciclaje no es sencillo y, según las características de los residuos, estos deben recibir tratamientos distintos para ser reciclados. Debido a este hecho,

los residuos han de ser estrictamente separados ya sea en origen o en las centrales de reciclaje. Sin embargo, como se ha demostrado en repetidas ocasiones, la separación de residuos ha de ser realizada tan pronto como sea posible para aumentar el valor de los materiales reciclados y permitir un reciclaje más eficiente. Por lo tanto, es en la separación en origen donde incide fundamentalmente el papel del consumidor, debiendo separar correctamente para que cada residuo siga el camino establecido.

En lo que respecta a Barcelona y sus alrededores, el sistema implementado mayormente es la recogida selectiva de cinco fracciones por deposición en contenedores disponibles en la vía urbana. En 2019 se aprobó un plan de gestión de residuos propuesto por el Área Metropolitana de Barcelona (AMB), llamado PREMETS con el siguiente objetivo:

El principal reto de la gestión de residuos en el área metropolitana es romper con la situación de estanqueidad de la recogida selectiva i alcanzar el 55% de reciclaje de residuos el año 2025 (MÉS I AMB MILLOR QUALITAT Recollir més i amb millor qualitat, s. f.).

Como ya se ha comentado, el papel del consumidor o ciudadano es de gran importancia ya que influye directamente en tres grandes claves para el éxito del sistema de reciclaje actual:

- Cantidad de ciudadanos participando activamente en la separación de los residuos para su reciclaje.
- Porcentaje de los residuos depositados frente a los generados.
- Calidad de los residuos, entendiéndose como la correcta deposición de estos en su correcta fracción.

Son numerosos y diversos los proyectos que se han diseñado y probado para facilitar al consumidor la separación de residuos en origen. Entre ellos el propuesto por Yann Glouche y Paul Coderc (Glouche et al., 2014), que consiste en un sistema de gestión de residuos inteligente completo. Este proyecto será descrito más adelante de forma más detallada.

Este trabajo pretende optimizar y *prototipar* una parte del sistema propuesto por Y. Glouche y P. Coderc, específicamente, la sección del contenedor doméstico de residuos inorgánicos que identifica el producto que se quiere reciclar y le recomienda al usuario la fracción donde debe ir depositado. Además, el objetivo es mejorar el dispositivo haciendo que participe en el *Internet of Things* o IoT y añadirle funcionalidades extra para hacer un producto más completo y atractivo. El dispositivo propuesto en este trabajo tiene como meta ayudar a resolver la problemática planteada por el PREMETS, promoviendo las tres claves mencionadas previamente.

3.2. Objetivos del proyecto

3.2.1. Objetivo principal

El objetivo principal del proyecto es el diseño de un contenedor doméstico de residuos inorgánicos que identifica el producto que se quiere depositar y recomienda al usuario la clasificación para su reciclaje.

3.2.2. Objetivos específicos

Del objetivo principal derivan los siguientes objetivos específicos:

- Analizar las alternativas tecnológicas de la identificación automática por radiofrecuencia y determinar las prestaciones y limitaciones que ofrece esta tecnología.
- Determinar las diferencias técnicas entre las distintas bandas frecuenciales y valorar la banda frecuencial que se adecue mejor a las necesidades del prototipo.
- Sopesar la viabilidad de un dispositivo final a partir de los resultados obtenidos con el prototipo.

La justificación para realizar el proyecto y alcanzar tanto la finalidad principal como las específicas de éste se basa en impulsar indirectamente los objetivos del siguiente árbol de objetivos:

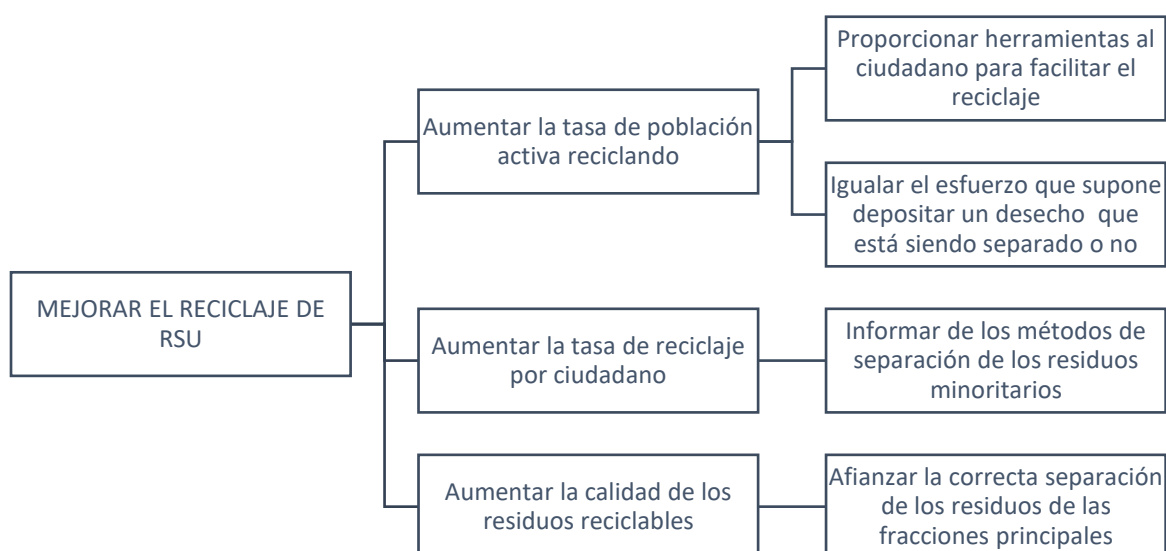


Figura 2. Árbol de objetivos

3.3. Alcance y estructura del proyecto

- En el capítulo de **introducción** se introduce el problema y su contexto y se justifica la realización del proyecto. Posteriormente, se definen los objetivos y la estructura del proyecto,
- En el capítulo de **estado del arte** se presentan proyectos previos relacionados con el proyecto, la tecnología de identificación automática y el Internet de las cosas. Además, se profundiza en la identificación automática por radiofrecuencia, introduciendo los elementos que participan en un sistema de identificación por radiofrecuencia, comparando la tecnología con el código de barras y presentando los estándares y legislación alrededor de esta tecnología.
- En el capítulo de **desarrollo de la idea** se describe el perfil de usuario objetivo para el que se diseñará el producto, se constatan los requisitos y funcionalidades para, posteriormente, introducir las diferentes tecnologías que podrían ser útiles para el desarrollo del contenedor.
- En el capítulo de **implementación del hardware** se justifica la elección de los componentes y se detallan sus aspectos técnicos. Posteriormente, se describe el montaje final de los distintos sistemas y los inconvenientes surgidos durante la implementación.
- En el capítulo de **implementación del software** se describe el *software* creado que se encarga del funcionamiento del dispositivo y se documentan las pruebas realizadas para determinar las prestaciones de cada sistema.
- En el capítulo de **valoración económica** se calcula el coste total del proyecto y la inversión realizada en el prototipo.
- En el capítulo de **impacto medioambiental** se resumen brevemente las condiciones y el proceso de reciclaje del prototipo.
- En el capítulo de **planificación del trabajo** se muestra la temporización del plan inicial y la duración y ejecución real del proyecto.
- En el capítulo de **conclusiones y líneas futuras** se exponen las conclusiones alcanzadas en el trabajo realizado, repasando los objetivos del mismo y los conocimientos adquiridos a lo largo de su realización. Finalmente, se proponen distintas líneas futuras o mejoras que se podrían adoptar en relación con el objeto del trabajo.

4. Estado del arte

En este cuarto apartado se exponen proyectos previos del ámbito medioambiental asociados a la tecnología de identificación por radiofrecuencia (en adelante, RFID, del inglés *Radio Frequency IDentification*) y se muestra la situación de las etiquetas RFID en los envases de alimentos. Posteriormente, se introduce la identificación automática y el Internet de las cosas y se profundiza en la identificación por radiofrecuencia.

4.1. Proyectos previos

Cada día aparecen más y más proyectos relacionados con la modernización y mejora del sector del reciclaje y es más habitual relacionar los conceptos de la preservación y mejora del medio ambiente y de la innovación tecnológica. En este sentido, se han desarrollado desde nuevos sistemas de recogida de residuos, como la recogida neumática, hasta nuevos contenedores urbanos que comprimen los desechos, pasando por contenedores domésticos que avisan de la cantidad de residuos generada. De los muchos proyectos realizados y en proceso de desarrollo cabe destacar los siguientes por su relación con este proyecto:

4.1.1. *A Smart Waste Management with Self-Describing objects*

Este estudio propone un sistema de gestión de residuos inteligente mediante la tecnología RFID, desde la separación en origen hasta la recogida colectiva de los residuos por parte de los servicios de gestión de residuos pertinentes. Para facilitar la viabilidad del sistema inteligente propuesto, éste se basa en el mismo esqueleto que nuestro sistema de gestión de residuos actual: el consumidor realiza la separación de residuos en contenedores domésticos para cada facción. Posteriormente, las bolsas de cada contenedor individual son depositadas en la entrada del edificio o en contenedores colectivos urbanos. Finalmente, los residuos colectivos son recogidos por los servicios de gestión de residuos respectivos. Por lo tanto, el objetivo de este proyecto es el de incluir en dicha estructura la tecnología RFID para beneficiarse de todas sus prestaciones y beneficios.

La renovación del sistema se inicia con la incorporación, por parte de los fabricantes, de etiquetas RFID en cada producto. Cada etiqueta RFID permite almacenar información como el contenedor al que debe ir depositado, el peso residual del producto, composición, etc. La información se puede introducir directamente en la etiqueta RFID o ésta puede contener un código asociado a información en un banco de datos externo. Independientemente de la solución adoptada, contenedores individuales con un lector RFID leen la información del *tag* e informan al usuario de en qué contenedor depositar el producto. Normalmente la deposición no se realiza directamente en el contenedor sino en una bolsa, para evitar ensuciar el contenedor y para facilitar el transporte de los residuos posteriormente. En consecuencia, Y.

Glouche y P. Coderc también plantean la posibilidad de incorporar una etiqueta RFID en la bolsa que permita identificar al propietario de ésta. Durante la inserción del residuo en la bolsa el lector RFID escribirá la información del producto en el *tag* de la bolsa. A continuación, cuando el usuario lo considera pertinente, traslada la bolsa al contenedor urbano colectivo o a los contenedores comunitarios del edificio. Cuando la bolsa es introducida en cualesquiera de los contenedores el código del propietario y el contenido queda registrado. Finalmente, la basura es recogida por los servicios de gestión de residuos, que pueden acceder a toda la información de los contenedores. Gracias a esta modernización del proceso de reciclaje el cliente puede obtener más información de la basura que genera y le facilita la separación en origen. Por otro lado, los servicios de gestión de residuos pueden identificar a quién pertenecen los residuos de cada bolsa y el contenido de la bolsa, y pueden monitorizar el flujo de residuos sólidos urbanos de manera más eficiente y sencilla.

Además, alternativamente, proponen un sistema inteligente de gestión de residuos con un enfoque más económico, mediante códigos QR. En este caso, cada residuo tiene la información asociada a un código QR. Para acceder a la información se utiliza un móvil inteligente con una aplicación que contiene un banco de datos. Con el móvil inteligente se escanea el código QR, obteniendo la información del banco de datos asociada a dicho código. Con este diseño alternativo se evita el coste del lector RFID, sacrificando comodidad y velocidad. El usuario debe escanear cada residuo para el correcto funcionamiento del sistema añadiendo un paso más en el proceso de reciclaje y una inversión de tiempo. En el documento se describe brevemente un prototipo basado en este último enfoque.

Aunque la propuesta de este Trabajo para desarrollar un proyecto de un contenedor doméstico inteligente se formuló inicialmente desconociendo la existencia del proyecto que se analiza en este apartado, la funcionalidad y el objetivo del contenedor individual que describen sus autores es, en esencia, el dispositivo que se quiere prototipar en este trabajo. Sin embargo, este estudio se centrará únicamente en el contenedor individual doméstico, analizando exhaustivamente sus aspectos técnicos, añadiendo más funcionalidades y optimizando su funcionamiento.

4.1.2. Dispositivos inteligentes: *Smart Trash Bin* y RFM

Son muchos los dispositivos que buscan aprovechar las nuevas tecnologías para aportar avances en el ámbito medioambiental.

Un diseño de un contenedor de basura inteligente para entornos urbanos, propuesto por Fady E. F. Samann (Albacete, 2006), es un claro ejemplo. El contenedor utiliza Arduino y un sensor de ultrasonido para monitorizar el nivel de llenado y alerta del estado del contenedor, enviando un SMS a los responsables competentes de la recogida de desechos. Una batería de litio acompañada de un panel solar alimenta el dispositivo.

De este proyecto es interesante extraer el uso de la placa Arduino y la función de monitorizar el nivel de llenado del contenedor. Sin embargo, el sistema está diseñado basándose en la estructura de gestión de residuos urbanos y no en la del entorno doméstico. Esta funcionalidad es bastante conocida y ha sido implementada en otros proyectos como *Smart Garbage Monitoring System for Waste Management* (Mohd Yusof, Jidin, & Rahim, 2017).

RFID *Food Management* (RFM) es un prototipo de un sistema inteligente que permite monitorizar los alimentos, cuyo envase contiene etiqueta RFID, presente en el frigorífico y la despensa del usuario (González, 2016). De nuevo, el sistema emplea Arduino conectado con un módulo RFID. Tiene como objetivo facilitar la gestión de alimentos en los hogares para evitar el desperdicio de comida y promover hábitos saludables.

Es de interés remarca que este proyecto parte del presupuesto de que los envases de los alimentos incluyan una etiqueta RFID. En el caso de RFM, la etiqueta RFID sirve para enviar información como la fecha de caducidad y/o información nutricional a una aplicación multiplataforma llamada RFM App. No obstante, el proyecto no realiza un estudio del arte de la situación actual de los envases con etiquetas RFID. Además, el prototipo también ha sido diseñado usando Arduino, lo que permite concluir que esta placa es adecuada para el diseño de prototipos similares como el planteado en el proyecto objeto del presente trabajo.

4.1.3. Etiquetas RFID en los productos alimentarios

La viabilidad del futuro desarrollo del contenedor doméstico inteligente planteado depende directamente de un factor externo al dispositivo: que el envase de los alimentos incluya una etiqueta RFID. Por lo tanto, es de gran relevancia indagar en la evolución y previsión de la incorporación de etiquetas RFID en los envases. Cabe destacar que ya se han realizado gran cantidad de estudios sobre los beneficios que aporta esta tecnología al sector y analizando las expectativas de su implementación en los próximos años.

Dado que los productores ofrecen productos de precio y calidad muy similares, en un entorno de fuerte competencia surge la necesidad de diferenciar sus productos. El envase se alza como elemento diferenciador, impulsando a los productores a aportar nuevas tecnologías y materiales (Clau, 2006). Una funcionalidad muy deseada por los productores es que el envase permita transmitir información, ya sea en la cadena de producción y suministro o al consumidor. Actualmente, dicha función es llevada a cabo por el código de barras. Sin embargo, esta tecnología está quedando obsoleta y la información que puede facilitar es limitada. Es por ello que las etiquetas RFID destacan como alternativa para ser incorporadas en los productos envasados.

Por otro lado, la tecnología RFID abre la puerta a desarrollar envases cada vez más inteligentes, con mayor número de funcionalidades. Un proyecto de 2015 desarrolló un

sistema RFID para la detección de la degradación de alimentos (Rodríguez, Del, & Alimentos, 2015). El sistema se centra en la trazabilidad del estado de alimentos que han de ser conservados a baja temperatura, concretamente la carne. Supone un avance respecto al cotidiano sistema de etiquetación de una fecha de caducidad orientativa que muchas veces tiende a ser errónea. Los resultados del proyecto fueron positivos y aseguran que la implementación del sistema permite discernir si el producto sigue en buen estado o no, en cualquier fase de la cadena de suministro.

4.1.4. PREMETS 2025

PREMET25 (Programa metropolitano de prevención y gestión de recursos y residuos municipales) es el proyecto impulsado por los 36 municipios del Área Metropolitana de Barcelona (AMB), con el apoyo de la Agencia de Residuos de Catalunya.

PREMET25 plantea un cambio en el sistema de recogida y tratamiento de residuos metropolitanos, no solo para llegar a cumplir los objetivos europeos, sino para incorporar una nueva lógica en la manera como se utilizan los recursos naturales i se gestionan los residuos («PREMET25 – Programa metropolità de prevenció i gestió de recursos i residus municipals / 2019-2025», s. f.).

PREMET25 nace con el objetivo de elevar hasta el 55% la cantidad de residuos reciclados frente a los residuos totales generados, ya que desde hace años la tasa de reciclaje de la zona se ha estancado en el 37%. Un porcentaje muy distante al impuesto por la Unión Europea. Además, entre otros muchos más objetivos, el proyecto pretende impulsar la transición hacia una economía circular, mejorar la cantidad y calidad de residuos recogidos selectivamente e incluso prevé el cierre de uno o dos de los tres hornos de la incineradora de Sant Adrià del Besòs, actualmente en funcionamiento. El programa prevé cumplir las expectativas en 2025.

Con dicho objetivo en mente, en varios municipios del territorio metropolitano se ha cambiado el sistema de recogida de residuos. En el nuevo sistema conocido como *Recogida puerta a puerta* se recogen los residuos en la puerta de cada casa según un calendario fijo. Los residuos orgánicos son recogidos entre 3 o 4 veces por semana, mientras las otras fracciones una única vez. Con la recogida puerta a puerta se eliminan la mayoría de contenedores urbanos, a excepción de unos pocos para emergencias. Se está analizando su funcionamiento y viabilidad para extenderlo a más municipios y, aunque está en fase de pruebas, sus primeros resultados son prometedores y optimistas. Resulta interesante, para los ciudadanos de estos municipios, que el contenedor doméstico pueda recordarle qué día y a qué hora se puede trasladar cada fracción de residuos desde el contenedor doméstico a la entrada del edificio.

4.2. Identificación automática

La identificación automática y captura de datos, AIDC (*Automatic Identification and Data Capture*) se refiere a los métodos empleados por dispositivos para identificar objetos automáticamente, recopilar datos de ellos e introducirlos directamente en sistemas informáticos.

Todas las tecnologías AIDC tienen el objetivo que se ha mencionado, pero se diferencian en la manera en que lo logran. Sin embargo, generalmente el dispositivo toma información externa, mediante imágenes, sonidos o vídeos, y la convierte en un archivo digital con la ayuda de un transductor. Es entonces cuando los datos pueden ser analizados, categorizados o tratados.

Los beneficios de utilizar sistemas AIDC son la reducción de costes, el control absoluto de los datos y la eliminación de los posibles errores en la introducción de información en el sistema. Además, se ahorra tiempo e, incluso, posiblemente personal, aumentando considerablemente la eficiencia del sistema. Entre los muchos usos de las tecnologías AIDC destacan la identificación de elementos, ya sea productos, personas o mascotas; la generación de inventarios automáticos y el control de acceso.

A continuación se explican brevemente algunas de las tecnologías AIDC más conocidas y destacadas:

- **CÓDIGO DE BARRAS:** es un sistema de codificación adaptado para ser leído rápidamente por un sistema óptico-electrónico. El código es creado a través de una secuencia de líneas (bandas) negras paralelas y espacios de distinto grosor.
- **TIRAS MAGNÉTICAS:** es un sistema de codificación formado por partículas ferromagnéticas incrustadas en una matriz de resina. La banda magnética es leída mediante contacto físico y gracias a la inducción magnética, pasándola a través de una cabeza lectora. Es conocida comúnmente como la banda oscura presente en las tarjetas magnéticas.
- **TARJETAS INTELIGENTES:** o tarjeta con circuito integrado (TCI), es un dispositivo físico, habitualmente, de plástico y del tamaño de una tarjeta de crédito. Incorpora un microprocesador capaz de almacenar y procesar información de forma segura.
- **SISTEMAS BIOMÉTRICOS:** sistema tecnológico que utiliza alguna característica biofísica para identificar un individuo. Cada persona posee una información biológica que lo diferencia de los demás y lo hace único. La identificación se realiza comparando una lectura con una referencia de la característica biofísica previamente registrada.

Algunos ejemplos de algunos rasgos biofísicos utilizados para identificar son la huella dactilar, la forma facial, el iris o la retina, el reconocimiento de voz, etc.

- RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR): es un proceso dirigido a la digitalización de textos a partir de una imagen. La imagen es analizada en busca de símbolos o caracteres pertenecientes a un alfabeto establecido, digitalizando la información para que pueda ser editada o almacenada.
- IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID): es un tipo de tecnología inalámbrica que permite la transmisión de información, mediante ondas de radio, entre un lector y una etiqueta, *tag* o transpondedor, que almacena la información.

4.3. Internet de las cosas

Internet de las cosas (de ahora en adelante IoT, del inglés *Internet of Things*) es la denominación que se otorga a la agrupación de dispositivos y objetos interconectados a través de una red. Los dispositivos que conforman el sistema pueden transmitirse información y/o interaccionar de máquina a máquina (*machine to machine* o M2M). Este nuevo paradigma de la computación proporciona un gran abanico de aplicaciones y prestaciones para mejorar la vida cotidiana, tanto en el entorno doméstico como en el entorno empresarial. Cualquier tipo de objeto o dispositivo puede participar en el IoT de las cosas y los más habituales son los sensores, dispositivos mecánicos y objetos cotidianos. Las aplicaciones son infinitas y actualmente ya hay ámbitos tecnológicos nacidos del IoT o sectores que lo han integrado. Ejemplos de este hecho son la domótica, los *wereables* inteligentes, la comunicación V2X (acrónimo del inglés *vehicle-to-everything*) y aplicaciones en monitorización de plantas de producción o salud de pacientes.

IoT es una tendencia tecnológica en desarrollo, que evoluciona continuamente y que tiene un futuro muy prometedor, con capacidad de cambiar drásticamente nuestro día a día. Sin embargo, como cualquier nueva tecnología, aún necesita madurar y existe cierta preocupación en ámbitos como la seguridad y la privacidad. El sector industrial del IoT deberá solventar dichos peligros para su correcta y extensa implementación.

4.4. Identificación por radiofrecuencia

La identificación por radiofrecuencia o RFID es un concepto que engloba todas las tecnologías que emplean las ondas de radio con el objetivo de identificar personas u objetos. La transmisión de la información, desde un *tag* o etiqueta a un lector, es inalámbrica sin necesidad de que exista contacto visual entre estos. En el momento que una etiqueta entra

en la zona de cobertura del lector, éste envía una señal para que dicha etiqueta le transmita la información que tiene almacenada en su memoria interna.

La tecnología RFID no es nueva en la industria y ya hay una fuerte y extendida implementación en ciertos sectores, como por ejemplo el pago con tarjetas *contactless*, el cobro a vehículos a distancia (*Teletac*), identificación de los empleados, etc. Aun así, sigue siendo una tecnología con un importante potencial de evolución y que día a día se va desarrollando y perfeccionando técnicamente. La introducción del RFID en nuevos sectores como la logística o la comercialización le dotan de gran relevancia y mayor diversidad.

4.4.1. Componentes de un sistema RFID

Los elementos que componen un sistema RFID, y que determinan sus características, son el tag, el lector, el *middleware*, la base de datos y el programador RFID. En términos generales, el funcionamiento de un sistema RFID básico es el siguiente: el lector envía una señal de radiofrecuencia con energía cada cierto tiempo, los *tags* responden transmitiendo la información que contienen. El lector recibe la información, que es tratada por un interfaz *middleware* para que sea almacenada en una base de datos de forma organizada y con un formato común. Gracias a este tratamiento previo, las aplicaciones clientes son capaces de trabajar y acceder a la información.

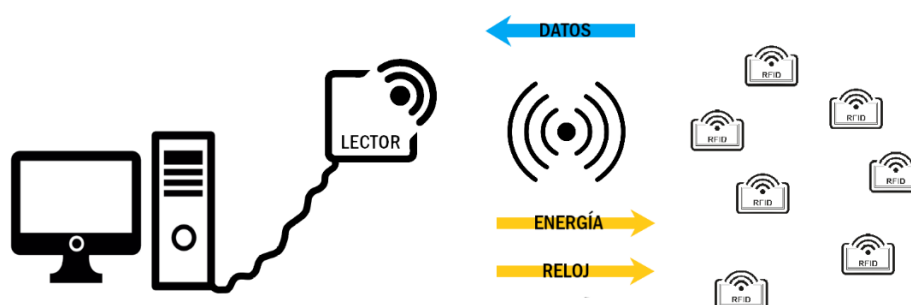


Imagen 1. Estructura de un sistema RFID.

Los fundamentos físicos de la radiofrecuencia, en los que se basa la comunicación entre los dispositivos RFID, generan diferentes modelos de transmisión según la banda de frecuencia en la que se esté realizando. Cada banda frecuencial tiene unas características específicas que determinan las prestaciones y limitaciones del sistema RFID. Las antenas, ya sean del lector o de las etiquetas, son los elementos del sistema RFID que realizan dicha comunicación radio frecuencial. Según su forma y características podrán trabajar en una banda frecuencial u otra. Para un correcto funcionamiento del sistema todas las antenas han de operar en la misma banda frecuencial ya que, en caso contrario, los dispositivos no podrán establecer comunicación.

Existen cuatro tipos de sistema RFID según su frecuencia de operación:

CLASIFICACIÓN	BANDAS FRECUENCIALES	CARACTERÍSTICAS	APLICACIONES
Baja Frecuencia (<i>LF</i> , Low Frequency)	30kHz – 300kHz, típicamente: 125kHz – 134,2kHz	Alcance: corto (hasta 45cm) Velocidad de transmisión: lenta Precio: bajo Interferencias causadas por líquidos: baja Interferencias causadas por metales: sí	Identificación de animales. Llave de automóviles. Control de accesos. Automatización en cadenas de producción.
Alta Frecuencia (<i>HF</i> , High Frequency)	3MHz – 30MHz, típicamente: 13,56MHz	Alcance: corto/medio (10cm – 1m) Velocidad de transmisión: lenta/moderada Precio: bajo/moderado Interferencias causadas por líquidos: baja Interferencias causadas por metales: sí	Ticketing. Pagos y transferencia de datos.
Ultra Alta Frecuencia (<i>UHF</i> , Ultra High Frequency)	3MHz – 2,4GHz, típicamente: 868MHz – 956MHz	Alcance: medio (3m – 10m) Velocidad de transmisión: moderada/rápida Precio: moderado/alto Interferencias causadas por líquidos: muy alta Interferencias causadas por metales: no	Almacenamiento y logística. Control de inventario.
Frecuencia de microondas	2,45GHz – 5,8GHz	Alcance: largo (más de 10m) Velocidad de transmisión: rápida Precio: alto Interferencias causadas por líquidos: muy alta Interferencias causadas por metales: no	Identificación de vehículos. Sistemas de peaje.

Tabla 1. Clasificación de los sistemas RFID según frecuencia de operación.

4.4.1.1. El tag

El *tag* o etiqueta RFID está compuesto por una antena, un transductor de radio y un microchip. El microchip tiene una memoria interna donde se almacena el identificador y, dependiendo de la etiqueta, información adicional. La función de la antena es recibir la señal de solicitud y la energía enviada por el lector y responder con la información almacenada en el microchip, después de que ésta haya sido convertida por el transductor.

Hay gran cantidad de *tags* RFID con características y prestaciones diferentes, que pueden ser clasificados según la fuente de energía y su frecuencia de operación. Estos dos criterios son clave para la caracterización del sistema RFID.

La alimentación del *tag* es una característica muy importante y determinante que es necesario tener en consideración, ya que influye en las prestaciones que otorga a la aplicación del sistema RFID. Además, establece el coste y la vida de la etiqueta. Atendiendo a esta clasificación, se pueden encontrar tres tipos de etiquetas: pasivas, activas o semi-pasivas.

- **Pasivas:** son las etiquetas más básicas y reciben la energía de la señal del lector. La ausencia de alimentación propia disminuye su coste y tamaño, y posee un ciclo de vida ilimitado. Su uso es el más extendido. No obstante, su funcionamiento depende del lector y la distancia de identificación se ve limitada a su rango.
- **Activas:** tienen alimentación propia, siendo más caras y de mayor tamaño. El ciclo de vida del *tag* viene determinado por el de su batería. También cuenta con transmisor propio, haciéndolo independiente a la señal del lector. Se utiliza cuando se quieren distancias de identificación elevadas.
- **Semi-pasivas:** tienen alimentación propia aunque, a diferencia de los *tags* activos, siguen dependiendo de la señal del lector. Igual que las etiquetas activas, su ciclo de vida está limitado al ciclo de vida de su batería.

4.4.1.2. El lector

El lector RFID, también llamado transceptor por su carácter transmisor/receptor, envía señales periódicamente y recibe las respuestas de las etiquetas que estén en sus inmediaciones. Para realizar la comunicación, el lector necesita una o varias antenas radio frecuenciales. Dependiendo del lector, puede tener la antena integrada en su *hardware* o puede estar conectado a antenas externas. Además, el lector incorpora un decodificador o procesador de señales para convertir y tratar la respuesta de los *tags* a un formato legible para las computadoras y el *middleware*. Como se ha mencionado, tanto el lector como su antena asociada han de ser seleccionados acorde a la banda frecuencial de operación deseada.

4.4.1.3. *Middleware* y base de datos

La información generada por un lector RFID no puede ser interpretada ni gestionada por una aplicación cliente. En consecuencia, es necesaria la intervención de una plataforma *middleware* para tratar los datos en bruto procedentes del lector. El *middleware* tiene principalmente tres responsabilidades: la coordinación de lectores, la entrega de datos y la gestión del proceso.

- Coordinación de lectores: configura y monitoriza los lectores dándoles comandos.
- Entrega de datos: asegura la precisión de los datos, filtra información redundante y la comprime.
- Gestión del proceso: el *middleware* puede analizar la información recibida y actuar en función de las reglas marcadas por la compañía. Por ejemplo, puede notificar al sistema la intrusión de un producto no autorizado o niveles bajos o rotura de stock.

La información de identificación es tratada para obtener un formato común determinado y almacenada en una base de datos. Finalmente, con la información almacenada de forma organizada y con un formato común, cualquier aplicación del cliente puede acceder a ella y gestionarla.

4.4.1.4. Programador RFID

El programador RFID es un dispositivo capaz de codificar ciertos datos en el microchip de las etiquetas. Dependiendo de las etiquetas, esta escritura se podrá realizar únicamente una vez, si el *tag* es de solo lectura o, por lo contrario, varias veces, si el *tag* es de escritura/lectura. Intuitivamente, se puede deducir que la escritura necesita mayor potencia que la lectura, requiriendo que la primera acción se realice en contacto directo entre el programador y el tag. El programador RFID es prescindible en los sistemas RFID donde es únicamente necesaria la lectura de etiquetas y, además, la mayoría de lectores RFID son capaces de escribir en un tag.

4.4.2. Comparativa entre la tecnología RFID y el código de barras

Habiendo descrito las características, prestaciones y limitaciones de la tecnología RFID, es interesante para este proyecto la comparativa entre ella y la del código de barras que se ha mencionado anteriormente. El objetivo de ambas tecnologías es la identificación automática, siendo el código de barras la primera tecnología que se implementó y la tecnología RFID un potencial sustituto de esta. Aún es pronto para saber si finalmente la tecnología RFID tendrá éxito frente al código de barras, pero está claro que presenta ciertas diferencias y ventajas en determinadas situaciones (Clau, 2006) (López, Sarrate, Suances, Teijeira, & Trigo, 2010), como son:

- Lectura más rápida y simultánea sin necesidad de visualización, dando la posibilidad de identificar los productos dentro de una caja o palé sin abrirlos, ahorrando tiempo y dinero.
- Almacenaje de mayor cantidad de información como los valores nutricionales, instrucciones de uso y reciclaje, información comercial o el estado del alimento. Además, se puede añadir o sobrescribir más información en la etiqueta con posterioridad a su incorporación en el producto.

Como cualquier tecnología en fase de implementación, todavía presenta una serie de inconvenientes que suponen desventajas frente al código de barras:

- Mayor coste de implantación y mantenimiento del sistema, coste de las etiquetas elevado respecto a la impresión de código de barras.
- Falta de estandarización internacional.
- Sistema vulnerable al espionaje y ataques técnicos.

Los problemas mencionados, fundamentalmente el coste, están frenando la implementación de las etiquetas RFID en los envases, pero se prevé que en muy pocos años la tecnología RFID esté tan implementada como lo está el código de barras (Clau, 2006). En la tabla 2 se muestran las diferencias de forma resumida.

CARACTERÍSTICAS	TECNOLOGÍA	
	CÓDIGO DE BARRAS	RFID
Alineamiento o aproximación	Requerido.	No requerido.
Artículos que pueden ser leídos	Uno.	Múltiples simultáneamente.
Automatización y precisión	Lectura manual y propenso a lecturas erróneas.	Totalmente automático y alta precisión.
Almacenamiento de datos	Solo los incluidos en el código. Datos fijos.	Más de varios Kbits. Se puede actualizar los datos.

Tabla 2. Comparativa del código de barras frente la tecnología RFID

4.4.3. Estándares y legislación

Como consecuencia de las diversas alternativas tecnológicas y variantes que puede adoptar un sistema RFID, era necesaria la implementación de estándares que permitieran la interoperabilidad entre aplicaciones. Actualmente ya existen unos estándares, aplicados con carácter internacional, que regulan las diversas áreas de los dispositivos RFID. La estandarización de la tecnología RFID se ha desarrollado principalmente en dos grupos de especificaciones: la Organización Internacional para la Estandarización o ISO y EPCglobal.

Hay cuatro áreas que la estandarización de la identificación RFID contempla:

- **PROTOCOLO DE INTERFAZ:** determina el modo y condiciones de la comunicación entre lector y etiquetas.
- **CONTENIDO DE DATOS:** detalla el formato y semántica de los datos transmitidos.
- **CERTIFICACIÓN:** pruebas establecidas para determinar si un dispositivo cumple los requisitos del estándar y garantizar la interoperabilidad entre dispositivos de diferente fabricante.
- **APLICACIONES:** usos comunes de la tecnología RFID.

4.4.3.1. GS1 - EPCglobal

GS1 es una organización mundial que nace de la fusión de EAN (*European Article Number*) y UCC (*Uniform Code Council*), y que tiene como objetivo el desarrollo y aplicación de estándares globales en las cadenas de abastecimiento (Aguirre, 2007). Las normas de GS1 permiten gestionar productos, servicios e información eficientemente y con seguridad y mejoran la trazabilidad de las cadenas de suministro y demanda. Sus estándares son ampliamente adoptados en todos los sectores y son los más utilizados a nivel mundial, tanto por grandes multinacionales como por pequeñas tiendas. Las ventajas de adoptar estos estándares son evitar la falsificación, y la reducción o rotura de stock (Ciáurriz, 2010). GS1 reúne los estándares globales en cuatro grupos diferentes:

- **GS1 Identify:** estándares que describen los diferentes identificadores para distinguir artículos, unidades logísticas, localizaciones o activos.
- **GS1 Capture:** estándares que incluyen las normas GS1 para los códigos de barras y los códigos electrónicos de producto (en adelante, EPC).
- **GS1 Share:** estándares centrados en la transmisión de información, las transacciones comerciales y la comunicación entre socios comerciales. Este conjunto incluye los estándares de *Global Data Synchronisation Network* (GSDN) y *EPC Information Systems* (EPCIS).
- **GS1 Use:** estándares para la trazabilidad de las cadenas de suministro.

EPCglobal, institución filial de GS1, es una de las dos organizaciones que ha desarrollado e impulsado estándares para el uso de la tecnología RFID. EPCglobal, por lo tanto, trabaja en los estándares del conjunto de GS1 Capture y su objetivo es incrementar la eficacia de las cadenas de suministro de todos los sectores industriales mediante el uso de la tecnología RFID y EPC. Las principales tareas del grupo son la creación de estándares y el asesoramiento y homologación de las diferentes aplicaciones RFID ya existentes. EPC Global ha definido estándares para cinco elementos distintos (Aguirre, 2007):

- **CÓDIGO ELECTRÓNICO DE PRODUCTO (EPC):** es el código numérico estandarizado que se asigna a un objeto para identificarlo de forma unívoca. El código no contiene información adicional y tiene cuatro atributos clave que posibilitan la identificación única de cada objeto (ver Figura 3):
 - **Encabezamiento** (*Header*), garantiza la singularidad del código EPC.
 - **Número de administrador del EPC** (*EPC Manager Number*), identifica la compañía o la organización propietaria del código.
 - **Clase del objeto** (*Object Class*), denomina la clase de producto.
 - **Número de serie** (*Serial number*), es el número de serie único para cada producto.

Los códigos EPC pueden estar compuestos desde 64 bits hasta 256 bits, aunque el formato más común y adoptado es el EPC de 96 bits, que permite generar una altísima cantidad de identificadores únicos.

96-bit EPC (formato decimal)					
48.	2.	5.	0078742.	135165.	000056710921
Header	Type	Partition	EPC Manager Number	Object Class	Serial Number
Asignado por EPCglobal				Asignado por el EPC Manager	

Figura 3. Ejemplo de código EPC de 96 bits en formato decimal.

- **PROTOCOLO DE COMUNICACIÓN:** el estándar EPCglobal otorga una clasificación de los *tags* y los lectores según sus prestaciones y capacidad. Esta clasificación estandarizada posibilita que las etiquetas y lectores puedan ser producidas por diferentes proveedores y, por lo tanto, el consumidor final no se vea obligado a comprarle todo el equipo al mismo vendedor. Este hecho impulsa la competencia en la industria y comporta una reducción de costes.
- **MIDDLEWARE EPC:** como se ha comentado anteriormente, el *middleware* es el elemento que conecta los lectores RFID con los sistemas de información de la compañía. Los estándares EPCglobal para el *middleware* RFID aseguran que la entrega de información sea eficiente y útil, evitando la transmisión de datos redundantes como dobles lecturas de la misma etiqueta o múltiples etiquetas.

- **SERVIDOR DE NOMBRE DE OBJETOS (ONS):** es una base de datos global que permite la búsqueda de información de un producto o servicios relacionadas a un EPC. El servidor ha sido diseñado basándose en el *Domain Name System* (DNS) para conseguir escalabilidad y funcionalidades similares. Cuando una etiqueta RFID es leída, el *middleware* envía el EPC al ONS, en una red local o Internet, para saber dónde se encuentra la información de dicho producto. Entonces, el ONS indica al *middleware* un servidor donde se almacena el archivo sobre ese producto. El *middleware* puede recuperar el archivo y enviarlo al inventario o aplicaciones de una empresa.
- **SERVICIO DE INFORMACIÓN EPC (EPCIS):** estándar que permite a los socios comerciales compartir información sobre el estatus de un producto a lo largo de la cadena de suministro. Se conforma de una base de datos central compartida que se actualiza continuamente por las aplicaciones que forman la red. Su objetivo es dar una visión compartida de los datos, tanto dentro como entre empresas, a diferentes aplicaciones de la cadena de suministro.

4.4.3.2. ISO

ISO, derivado del término griego *isos* que significa igual, es una organización internacional no gubernamental que se encarga de desarrollar estándares y normas internacionales de fabricación y especificaciones de productos, servicios y sistemas. Los estándares aseguran que los productos sean iguales, en calidad y eficiencia, sea cual sea el país donde se ha fabricado. Está formada por 164 miembros, siendo cada miembro la principal organización de normalización de su país. Cada país solo puede tener un único representante y las compañías y los particulares no pueden ser miembros de la ISO. En la tabla número 3 se resume brevemente algunos de los estándares ISO relacionados con la tecnología RFID:

DENOMINACIÓN	ASUNTO
ISO/IEC 15961	Define el protocolo de datos e interfaz de aplicación para gestión de objetos.
ISO/IEC 15962	Fija el protocolo de codificación de datos y funcionalidades de la memoria para gestión de objetos.
ISO/IEC 15963	Trata el sistema de trazado y monitorización que afecta al <i>tag</i> RFID para gestión de objetos.
ISO/IEC 19762-3	Define el vocabulario de la tecnología RFID dentro de las técnicas AIDC.
ISO/IEC 18000	Define el interfaz aéreo para dispositivos RFID LF, HF, UHF y frecuencia de microondas.
ISO/IEC 18046	Determina los métodos de prueba para testear el rendimiento del sistema, lectores o tags.

ISO/IEC 18047	Describe los métodos de prueba para testear las comunicaciones de interfaz aérea.
ISO/IEC 24729	Especifica las pautas para la implementación de la tecnología RFID, trata aspectos como el reciclaje de etiquetas y la instalación de antenas o lectores.
ISO/IEC 24791	Detalla la infraestructura del sistema de <i>software</i> .

Tabla 3. Lista de ISO relacionadas con la tecnología RFID

4.4.3.3. Legislación

Además de los estándares EPCglobal e ISO, hay que tomar en consideración otras normas que afectan directamente a la tecnología RFID. El Instituto Europeo de Normas de Telecomunicación (en adelante, ETSI) se encarga de producir normas, con proyección mundial, sobre la industria de las telecomunicaciones. Algunas de estas normativas incumben la propagación por ondas de radio y por tanto repercuten en la tecnología RFID:

DENOMINACIÓN	ASUNTO
EN 300 220	Dispositivos de corto alcance en el rango de 25MHz – 1000MHz: Características técnicas y métodos de medición. Relevante para RFID en UHF.
EN 300 330	Equipos de radio en el rango de frecuencia de 9kHz – 25MHz y sistemas inductivos en el rango de frecuencia de 9kHz – 30MHz. Relevante para RFID en LF y HF.
EN 300 440	Equipos de radio para usarse en el rango de frecuencia de 1GHz – 40GHz. Relevante para RFID en frecuencia de microondas.
EN 300 674	Equipos de transmisión DSRC operando la banda de 5,8 GHz. Relevante para RFID en frecuencia de microondas.
EN 302 208	Equipos de identificación por radiofrecuencia, operando en la banda 865MHz - 868MHz de hasta 2 W de potencia y, operando en la banda 915MHz – 921MHz de hasta 4 W de potencia. Relevante para RFID en UHF.
EN 302 328	Sistemas de transmisión de banda ancha. Relevante para RFID en frecuencia de microondas.

Tabla 4. Lista de normas ETSI relacionadas con la tecnología RFID.

5. Desarrollo de la idea

En este capítulo se describirá el perfil de usuario objetivo para el que se diseñará el producto, se constatarán los requisitos y funcionalidades del prototipo y se describirá cómo se espera que funcione. Finalmente, se introducirán las diferentes opciones de *hardware* que podrían ser útiles para el desarrollo del contenedor.

5.1. Definición del usuario

Para el buen desarrollo de cualquier proyecto de prototipado, el usuario final es quien, indirectamente, determina qué características y funcionalidades han de ser priorizadas. Es necesario, entonces, saber las necesidades del consumidor y lo que éste espera del producto para satisfacer sus expectativas y cumplir los requisitos que considere esenciales en el prototipo.

Sobre la base de este razonamiento se ha realizado una breve encuesta para definir al usuario y los requisitos y funcionalidades que se requerirán al prototipo. La encuesta se ha realizado a ciudadanos residentes en Barcelona y mediante la herramienta de *Google Drive: Formularios de Google*. Además, se ha diseñado la encuesta con un fin didáctico, evitando que su respuesta sea una tarea difícil y tediosa. Los resultados de la encuesta figuran en el anexo 1.

En este proyecto, para este análisis, se han considerado dos rasgos distintivos: la edad y si la persona practica o no el reciclaje. En relación con el usuario, de la encuesta se pueden extraer las siguientes conclusiones:

- Un 89,9% de los encuestados recicla y que lo haga o no es independiente de la edad; tanto mayores como jóvenes practican el reciclaje.
- Los encuestados que no reciclan (10,1%) afirman que ello se debe, en primer lugar, a la falta de espacio en casa y, en segunda lugar, a que los contenedores están lejos de su hogar.
- El 50,6% de los encuestados percibe correctamente la cantidad de residuos que genera una persona al día. El hecho de que la persona recicle o no lo haga no es estadísticamente relevante.
- Un 60,8% de los encuestados percibe correctamente el porcentaje de residuos que se reciclan en el área metropolitana de Barcelona. El hecho de que la persona recicle o no lo haga no es estadísticamente relevante.
- El porcentaje de respuestas correctas a todas las preguntas de clasificación de residuos es del 40,1%. Cabe destacar que, además, ninguno de los encuestados ha respondido correctamente a todas las preguntas de clasificación de residuos. El

porcentaje de respuestas correctas de los encuestados que reciclan es ligeramente superior (39,7%) frente a los que no reciclan en el hogar (35,4%).

- Las estadísticas muestran que el 89,9% de los encuestados confían haber marcado más de la mitad de las respuestas correctamente.
- Hay interés por un contenedor doméstico inteligente tanto por los encuestados que reciclan (60,5%) como por los que no (75%).
- Los encuestados se muestran muy interesados en un contenedor doméstico inteligente que les indique en qué fracción depositar el residuo que están desechando (73,4%). El interés disminuye ligeramente respecto a la función de indicar la cantidad de residuos generada y separada (70,9%), aunque sigue siendo elevado.
- La media del precio máximo que estarían dispuestos a pagar los encuestados está entre los 20€ y los 40€.

Los datos evidencian que hay un claro desconocimiento de cómo se debe separar los residuos en el hogar. Si se extrapolan estos resultados a la población de una ciudad, prácticamente la mitad de la población no es consciente de la cantidad de residuos que genera y desconoce el porcentaje de residuos reciclados en su ciudad. Está claro, por lo tanto, que aún se debe impulsar la concienciación y la participación de la ciudadanía en el reciclaje. Además, prácticamente tres cuartas partes de los encuestados muestran interés por una herramienta que les ayude a reciclar, lo que indica que hay voluntad de contribuir a la mejora del medioambiente y de gestionar mejor los residuos.

Por ende, está plenamente justificada la necesidad de dar herramientas que ayuden a la ciudadanía para que desempeñe el reciclaje. El usuario principal al que irá dirigido el contenedor doméstico inteligente son las personas que actualmente ya reciclan y separan residuos, sin importar la edad. No obstante, si se llegase a reducir el espacio necesario para reciclar residuos, las personas que por el momento no reciclan podrían convertirse en potenciales consumidores del producto, ya que el espacio es el principal motivo por el cual no reciclan.

5.2. Definición de requisitos y funcionalidades

Un proceso clave en el diseño de un prototipo es discernir qué requisitos y funcionalidades debe tener para maximizar la satisfacción del consumidor. El usuario puede perder el interés si el producto no tiene todas las funcionalidades que esperaba o generar decepción si no se cumplen los requisitos deseados. Del cuestionario realizado, referente a los requisitos y funcionalidades, se han constatado los siguientes hechos:

- Las funcionalidades de indicar en qué fracción depositar el desecho e indicar la cantidad de residuos generada son atractivas para el consumidor final.

- El precio de venta del contenedor doméstico inteligente ha de ser de entre 20 y 40€.

Sin embargo, dado los costes relacionados con la tecnología RFID, este proyecto priorizará diseñar un prototipo que proporcione las dos funcionalidades deseadas por el usuario definido. Por ello, aunque también se hará una valoración económica del prototipo y se calculará el coste, éste no se plantea como una limitación a la hora de definir el prototipo. La minimización de costes y precio de venta se dejará para líneas futuras.

Por lo tanto, añadiendo los requisitos de un contenedor doméstico clásico, el prototipo deberá cumplir con las siguientes características y funcionalidades:

CARACTERÍSTICAS FÍSICAS

- Contenedor con capacidad de 15L.
- Contenedor impermeable.
- Contenedor con tapa.
- Conexión con alimentación eléctrica de 220V.

FUNCIONALIDADES

- Identificación automática inalámbrica a distancia.
- Identificación simultánea de objetos.
- Sistema de aviso de llenado mediante sensor de ultrasonido.
- Conexión y configuración inicial a Internet.

Figura 4. Lista de características y funcionalidades del prototipo

5.3. Alternativas de diseño

En este apartado se describirá el planteamiento inicial del sistema y cómo el dispositivo interactúa con el usuario.

La funcionalidad principal diferencial de este dispositivo inteligente frente a los cubos de residuos convencionales es que identificará el producto que se quiere depositar y recomendará al usuario en qué fracción ha de ser introducido según el reglamento medioambiental vigente. Para ello dispondrá de un sistema RFID muy sencillo, formado únicamente por un lector, incorporado en el contenedor, y etiquetas RFID, incorporadas en los productos a depositar. Como se ha descrito en apartados anteriores, los *tags* RFID contienen, como mínimo, un código de identificación. Hay etiquetas que pueden guardar información extra. El usuario, al acercarse al dispositivo con el residuo, introducirá el *tag* RFID dentro del rango de lectura del lector. Entonces, el lector extraerá la información contenida en la etiqueta y la enviará al programa *software*. El programa *software* tratará la información recibida y, si se ha recibido la fracción del producto, enviará una determinada orden al sistema

de visualización. Si únicamente se recibe el identificador, será necesario compararlo con una base de datos que incluya la información del producto y entonces transmitirla al sistema de visualización. Al recibir la orden, el sistema de visualización simplemente informará al usuario de la fracción donde depositar el residuo. El usuario depositará el residuo según la recomendación del dispositivo.

Paralelamente, un sistema de medición de residuos generados o de monitorización del nivel de llenado transmitirá continuamente la información captada por sus sensores al procesador. El procesador realizará diversos cálculos y enviará el resultado al sistema de visualización.

Finalmente, un sistema de conectividad a Internet permitirá configurar la conexión inicial y mantendrá la conexión a la red.

5.4. Alternativas tecnológicas

Atendiendo a la introducción de la tecnología RFID y según el planteamiento del sistema descrito, se presentarán las opciones tecnológicas aplicables a este proyecto.

5.4.1. Placa de desarrollo

En primer lugar, a fin de diseñar un prototipo funcional es necesario que éste incorpore un componente que controle todos los elementos del dispositivo. Existen dos alternativas tecnológicas muy populares y económicas que satisfacen dicha función: Arduino y Raspberry Pi. Ambas plataformas son adecuadas y asequibles para realizar proyectos de electrónica. Sin embargo, son dos productos que fueron diseñados para cosas distintas y, por lo tanto, presentan una serie de diferencias, teniendo cada una ventajas y desventajas respecto a la otra. A continuación se describen brevemente ambas opciones para valorar cuál se ajusta mejor a las necesidades del proyecto.

Arduino es una plataforma de desarrollo de electrónica flexible, sencilla y de *hardware* y *software* libres. Gracias a la combinación de ambos elementos, *hardware* y *software* Arduino, es posible crear todo tipo de microordenadores de una sola placa. El *hardware* de Arduino consiste en una placa basada en un microcontrolador Atmel. El microcontrolador es un circuito integrado programable en el que se graban instrucciones. El microcontrolador tiene una interfaz de entrada y una de salida para conectar la placa con los periféricos. La interfaz de entrada se encarga de transmitir la información de los periféricos al microcontrolador mientras la interfaz de salida envía información procesada o órdenes a otros periféricos.

Existen varios modelos de placas Arduino con diferentes características y prestaciones. La siguiente tabla presenta un resumen de las características de los modelos más populares y que se han diseñado para adecuarse a todo tipo de proyectos:

MODELO	UNO	NANO	MEGA	ZERO
Microcontrolador	ATmega328P	ATmega328	ATmega2560	ATSAMD21G18
Voltaje operativo	5 V	5 V	5 V	3,3 V
Voltaje de entrada recomendado	7-12 V	7-12 V	7-12 V	3,3 V
Pines digitales de Entradas/Salidas	14, 6 de ellos salidas PWM	22, 6 de ellos salidas PWM	54, 15 de ellos salidas PWM	20, 10 de ellos salidas PWM
Pines analógicos de entrada	6	8	6	6
Memoria Flash	32 kB	32 kB	256 kB	256 kB
SRAM	2 kB	2 kB	8 kB	32 kB
EEPROM	1 kB	1 kB	4 kB	None
Frecuencia de reloj	16 MHz	16 MHz	16 MHz	48 MHz

Tabla 5. Características generales de los modelos principales de placas Arduino.

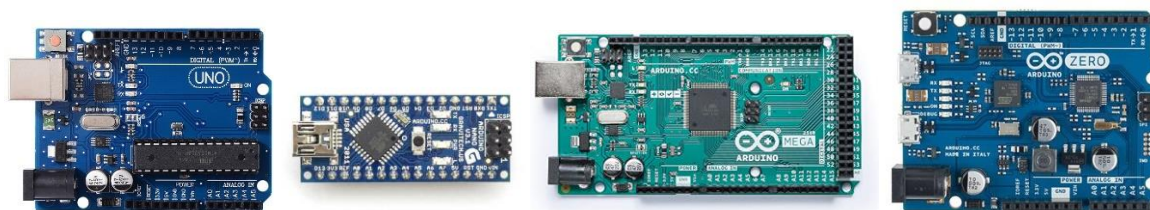


Imagen 2. Placa UNO, Nano, Mega y Zero respectivamente.

Existen muchos modelos más, pero el resto ya tienen usos más concretos. Para ver la lista completa de placas Arduino oficiales acceder a («Boards & Modules - Arduino», s. f.).



Imagen 3. Otros modelos de placas Arduino diseñados para usos más concretos.

Referente al *software* de Arduino, la plataforma ofrece el entorno de desarrollo integrado de Arduino (en adelante, IDE, del inglés *Integrated Development Environment*), un entorno de programación que permite a cualquiera crear aplicaciones para las placas. Desde el IDE se puede escribir el programa con una versión simplificada del lenguaje de programación C++, compilarlo y posteriormente volcarlo al microcontrolador.

Con la plataforma Arduino se pueden desarrollar una inmensa cantidad de proyectos. Algunos ejemplos son una báscula digital, un lector de huellas para la puerta del garaje, un control remoto universal, persianas controladas por voz, etc.

Por otro lado, Raspberry Pi es una computadora de placa completamente funcional, simple y económica. Su sistema operativo también es de código abierto, al contrario que su *hardware*; el fabricante mantiene el control sobre el diseño y la fabricación de las placas. En Raspberry Pi hay dos familias de placas: la familia “principal” y una familia de placas de tamaño más reducido llamado “Zero”. La gama principal de placas Arduino tiene 4 modelos y únicamente los modelos Raspberry Pi 1 y 3 tuvieron más de una versión: A, B y B+. Los demás modelos principales solo tienen la versión B. Las versiones B de cada modelo son las más populares

y habituales en la comunidad *Maker*.

MODELO	Raspberry Pi 1 B+	Raspberry Pi 2 B	Raspberry Pi 3 B	Raspberry Pi Zero	Raspberry Pi Zero W
SOC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2835	Broadcom BCM2835
Core	ARM1176JZF-S	Cortex-A7	Cortex-A53 64-bit	ARM1176JZF-S	ARM1176JZF-S
Nº Cores	1	4	4	1	1
Frecuencia de reloj	700 MHz	900 MHz	1.2 GHz	1 GHz	1 GHz
RAM	512 MB	1 GB	1 GB	512 MB	512 MB
Memoria	Micro SD	Micro SD	Micro SD	Micro SD	Micro SD
USB	2	4	4	1 microUSB	1 microUSB
Ethernet	SI	SI	SI	No	No
Wi-Fi	No	No	SI	No	SI
Bluetooth	No	No	SI	No	SI
HDMI	Si	Si	Si	Mini	Mini

Tabla 6. Características generales de los distintos modelos de placas Raspberry Pi.

Las placas Raspberry Pi tienen su propio sistema operativo oficial llamado Raspbian. Raspbian es una distribución del sistema GNU/Linux, concretamente una versión de la distribución Debian, y fue diseñado para la enseñanza de informática utilizando las placas. Sin embargo, también se pueden instalar otros tipos de sistemas operativos.

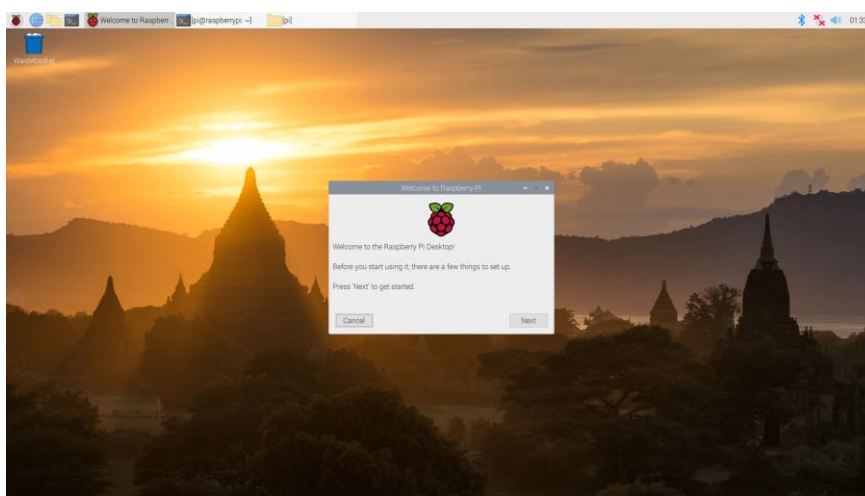


Imagen 4. Pantalla principal del SO Raspbian

Al igual que con Arduino, se pueden crear una infinidad de proyectos basados en Raspberry Pi. Algunos de ellos son un altavoz inteligente, tu propio servidor VPN, sistema automático de bombeo de agua, un sistema central de una casa inteligente, etc.

Habiendo hecho una breve explicación de ambas opciones, procede señalar las diferencias entre ellas. Éstas pueden ser cruciales para decantarse en la elección de una de las alternativas frente a la otra, según las necesidades del proyecto.

Como se ha mencionado anteriormente, la primera diferencia que destaca es la filosofía del *hardware*. Arduino da la posibilidad de que cualquiera pueda crear sus propias versiones de la placa, mientras que, por el contrario, los modelos de Raspberry Pi están limitados a los que ofrece el fabricante. Esto puede ser esencial si se quiere realizar un proyecto que requiera una placa con prestaciones inusuales o muy variadas. Por ello, cabe la posibilidad de que ninguna de las placas en el mercado satisfaga los requisitos del proyecto y sea necesario customizar la placa según el criterio del usuario.

En segundo lugar, las placas Arduino cuentan con entradas tanto analógicas como digitales, mientras que las placas Raspberry Pi tienen únicamente entradas digitales. Se debe principalmente a que Raspberry se diseñó como un ordenador en sí y, en cambio, Arduino se ideó como una herramienta para que cualquiera pueda crear el proyecto que desee. Además, activar y desactivar las entradas con Arduino es mucho más sencillo y rápido y las placas Arduino ejecutan inmediatamente el programa volcado al enchufarse. Raspberry Pi requiere de un sistema operativo completo, por lo que su inicio es más lento.

No obstante, la Raspberry Pi, al ser diseñada como un ordenador, tiene más potencia de procesamiento y cálculo que las placas Arduino. La capacidad de las placas Arduino podría no ser suficiente si en el proyecto fuera necesario realizar un procesamiento más elaborado o complejo. Finalmente, las placas Raspberry Pi cuentan con conectividad WiFi y Ethernet integradas. Para habilitar la misma conectividad en las placas Arduino es necesario añadirle una placa de expansión o microchip, utilizando parte de sus puertos.

En definitiva, Arduino es una alternativa más versátil, sencilla y rápida, pero con una potencia de procesamiento moderada y requiere de elementos externos para tener conectividad WiFi y/o Ethernet. Al contrario, Raspberry Pi cuenta con la conectividad ya integrada y una mayor potencia de cálculo.

5.4.2. Sistema RFID

Como se ha descrito previamente, la tecnología RFID presenta diversas alternativas tecnológicas según la frecuencia de trabajo. Teniendo en cuenta los aspectos tecnológicos, tres de las cuatro bandas frecuenciales típicas del RFID son adecuadas para construir el prototipo: la banda de baja frecuencia (LF), la de alta frecuencia (HF) y la de ultra alta

frecuencia (UHF). Se descarta la banda frecuencial de microondas por motivos que se explicarán más adelante en esta sección. Aunque las tres bandas frecuenciales son convenientes, cada una tiene unas prestaciones y limitaciones que influirán directamente en el funcionamiento del contenedor.

Actualmente, la tecnología RFID está muy presente en la vida cotidiana de gran parte de la población como, por ejemplo, el uso de la tecnología inalámbrica de corto alcance (en adelante, NFC, del inglés *Near-Field Communication*) como método de pago o para acceder al transporte público, ya sea con la tarjeta o con el móvil, y los sistemas de identificación automática por tarjeta en los lugares de trabajo. La tecnología RFID se ha convertido en una opción muy atractiva y muy popular para realizar proyectos electrónicos propios entre la comunidad *Maker*. En consecuencia y/o de la mano del creciente interés por el RFID, la industria electrónica ha desarrollado y ofrece varias alternativas compatibles con Arduino y Raspberry Pi para cada una de las bandas, tanto de módulos o *shields* lectores como de *tags* RFID. Sin embargo, hay una gran diferencia entre las opciones disponibles de LF y HF frente a las de UHF. Hay una considerable variedad de componentes de LF y HF en el mercado y, además, muy asequibles. En cambio, en el caso de la banda UHF las opciones son limitadas y con un precio muy elevado. A medida que la frecuencia de trabajo de los componentes aumenta, su precio también aumenta, incluso de LF a HF. Pero el salto de precio entre HF y UHF es significativamente muy superior, especialmente en los lectores.

Finalmente, se ha descartado la banda frecuencial de microondas por las siguientes razones. En primer lugar, las prestaciones de esta banda frecuencial exceden las funcionalidades y requisitos del contenedor. Con el fin de diseñar un prototipo económicamente viable y eficiente no es necesario implementar un sistema RFID que opere en la frecuencia de microondas. En segundo lugar, no hay lectores RFID de la banda frecuencial de microondas compatibles con Arduino y Raspberry Pi disponibles en el mercado. Además, los lectores no compatibles que se ofertan tienen un precio muy elevado y un tamaño considerable, características ambas inapropiadas para la implementación de un prototipo.

Tras la justificación del descarte de la banda de microondas, a continuación se presentan las alternativas tecnológicas para cada una de las tres bandas frecuenciales: LF, HF y UHF.

5.4.2.1. LF

La frecuencia de operación típica de casi la totalidad de lectores y etiquetas RFID de LF es de 125kHz. En la siguiente tabla figuran los lectores más populares acompañados de sus características generales:

PARÁMETRO	MÓDULO			
	7941E	RDM630	RDM6300	ID-12LA
Frecuencia de trabajo	125kHz	125kHz	125kHz	125kHz
Velocidad de transmisión	9600	9600	9600	9600
Interfaz	Wiegand, RS232	Wiegand, RS232	Wiegand, RS232	UART, RS232
Etiquetas compatibles	EM4100	EM4100	EM4100	EM4100
Voltaje de trabajo	4,75-5,25V	5V	5V	2,8-5V
Corriente de consumo	40mA	<50mA	<50mA	35mA
Distancia de lectura	0-70mm	20-50mm	20-50mm	0-120mm
Tamaño del PCB	47x26x5mm	38.5x19x9mm	38.5x19x9mm	25x26mm

Tabla 7. Características generales de los módulos lectores RFID de LF: 7941E, RDM630, RDM6300 e ID-12LA.

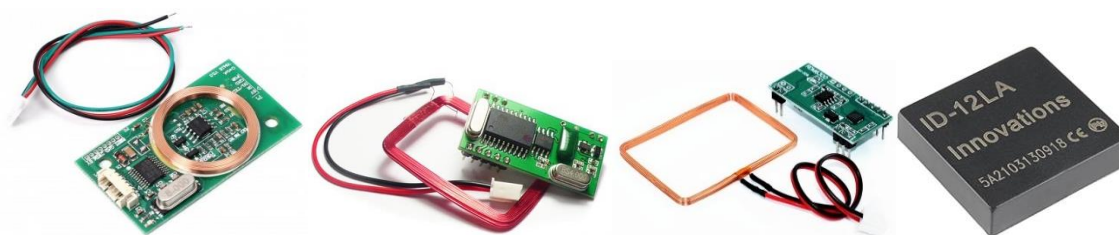


Imagen 5. Módulos lectores RFID de LF: 7941E, RDM630, RDM6300 e ID-12LA, respectivamente.

En la banda de baja frecuencia hay muchos modelos distintos de tarjetas RFID muy económicas. Sin embargo, las etiquetas RFID que predominan en la industria integran el chip EM4100 o el chip T5577. EM4100 es un chip sencillo de solo lectura y con una capacidad de 64 bits. Por lo tanto, las etiquetas RFID EM4100 solamente almacenan un identificador pregrabado de fábrica no reescribible. En cambio, el chip Atmel Temic T5577 es un chip de lectura/escritura que tiene una memoria de 363 bits, de los cuales 264 bits están disponibles para grabar información adicional.

5.4.2.2. HF

La frecuencia de operación típica de casi la totalidad de lectores y etiquetas RFID de HF es de 13,56MHz. A continuación se presentan los módulos lectores más destacados, compatibles con Arduino y Raspberry Pi, de la banda de alta frecuencia:

PARÁMETRO	MÓDULO			
	MFRC522	ElecFreaks	Adafruit	SM5210
Frecuencia de trabajo	13,56MHz	13,56MHz	13,56MHz	13,56MHz
Interfaz	SPI, UART, I2C	SPI, I2C, UART	SPI, I2C, UART	UART, I2C
Etiquetas compatibles	MIFARE, NTAG	MIFARE, Felica	MIFARE, Felica	MIFARE
Voltaje de trabajo	3,3V	5V	3,3V	3,3 - 5V
Corriente de consumo	10-13mA	<90mA	75mA	180mA
Distancia de lectura/escritura	0-50mm	20-50mm	100mm	80mm
Tamaño del PCB	85x54 mm	85x52mm	110x150mm	27x22mm

Tabla 8. Características generales de los módulos lectores RFID de HF: MFRC522, módulo NFC de elecFreaks, módulo NFC de Adafruit y SMS5210.



Imagen 6. Módulos lectores RFID de HF: MFRC522, módulo NFC de elecFreaks, módulo NFC de Adafruit y SMS5210.

Existen dos estándares ISO para el formato de los datos contenidos en las etiquetas de HF: ISO15693 y ISO14443. El primer estándar, también conocido como “*Vicinity*”, garantiza lecturas hasta unos 1,5 metros, mientras las lecturas del segundo, conocido como “*Proximity*”, alcanzan hasta los 12cm. En la ISO14443 se incluyen los *tags* MIFARE, las etiquetas de uso más habitual en todo el mundo. Hay diferentes modelos de etiquetas MIFARE, agrupadas en las siguientes gamas:

- *MIFARE Classic*: gama diseñada para el almacenamiento de memoria con seguridad. Incluye dos modelos, 1K (768 bytes) y 4K (3kB).
- *MIFARE Ultralight*: etiqueta de 64 bytes sin seguridad, utilizada a menudo de forma

desechable.

- *MIFARE T=CL*: tarjetas que incorporan un sistema operativo para poder realizar operaciones complejas rápidamente y con seguridad.
- *MIFARE DESfire*: similar a la *MIFARE Classic 4k*, pero con un *software* de propósito general incorporado. Es más flexible, tiene una mayor seguridad y mayor rapidez.

El modelo más vendido y utilizado son las *MIFARE Classic 1K*.

5.4.2.3. UHF

Al contrario que las otras dos bandas frecuenciales, la frecuencia de operación típica de la banda de ultra alta frecuencia no es la misma en todo el mundo. Sin embargo los lectores UHF trabajan siempre a una frecuencia entre 840MHz y 960MHz.

PARÁMETRO	MÓDULO	
	Sparkfun M6E-NANO	Cottonwood UHF Reader
Frecuencia de trabajo	840 - 960MHz	840 - 960MHz
Interfaz	FTDI, UART	UART, USB
Etiquetas compatibles	ISO18000-6C (EPC gen2)	ISO18000-6C (EPC gen2)
Voltaje de trabajo	5V	5V
Corriente de consumo	1A	2A
Distancia de lectura/escritura	Hasta 5m	1 - 6m
Tamaño del PCB	78x53mm	90x49mm

Tabla 9. Características generales de los módulos lectores RFID de UHF: Sparkfun M6E-NANO y Cottonwood.

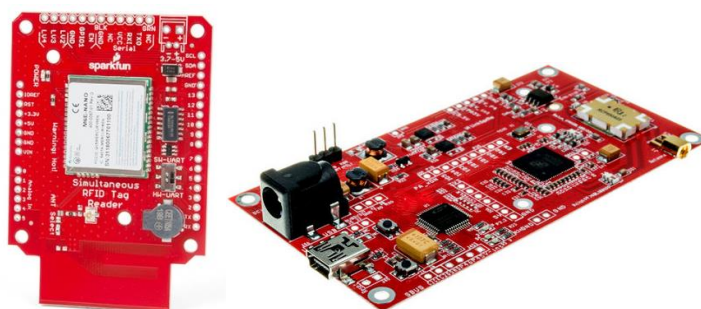


Imagen 7. Módulos lectores RFID de UHF: Sparkfun M6E-NANO y Cottonwood.

5.4.3. Cantidad de residuos generados y nivel de llenado

En el estudio de usuario se ha determinado que una función del contenedor muy atractiva para el consumidor es que éste monitoree e informe de la cantidad de residuos generados o del nivel de llenado. La cantidad de residuos generados se puede calcular según el peso de los residuos introducidos, mientras el nivel de llenado se puede medir según el volumen que ocupan dichos desechos. Más adelante se analizará si implementar ambas prestaciones o únicamente una de las dos.

La opción tecnológica más práctica para determinar el peso de los residuos son las células de carga. Las células de carga, o sensores de fuerza, son transductores que envían una señal eléctrica proporcional a la fuerza que se les aplica. Las básculas digitales se basan habitualmente en el uso de este componente. En la tabla 10 se muestran algunas de las células de carga compatibles con Arduino y Raspberry Pi, junto con sus características generales.

PARÁMETRO	TAS606	SPRK-SEN-09376	YZC-133
Rango de peso	5kg a 500kg	100 g a 10 kg	Hasta 5kg
Longitud	2,0 cm	8,9 cm	7,5 cm
Anchura	1,1 cm	4,5 cm	1,3 cm
Área sensora	Según la plataforma incorporada	4,5 cm x 3,8 cm	Según la plataforma incorporada

Tabla 10. Características generales de las células de carga: TAS606, SPRK-SEN-09376 e YZC-133.

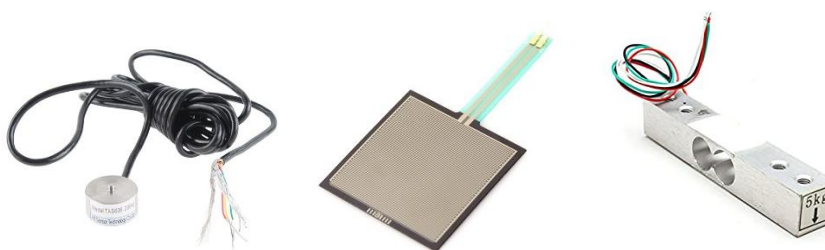


Imagen 8. Células de carga TAS606, SPRK-SEN-09376 e YZC-133, respectivamente.

Dependiendo del modelo de célula de carga empleado, es también necesario utilizar un amplificador de célula de carga y una plataforma. Esto se debe a que las lecturas de algunas células de carga pueden ser complicadas de tratar directamente con el Arduino o Raspberry Pi. El amplificador facilita la lectura, convirtiendo la señal analógica leída a una señal digital y amplificando dicha señal. El amplificador de célula de carga más popular y utilizado es el modelo HX711.



Imagen 9. Amplificador de célula de carga HX711.

Por otro lado, para determinar el nivel de llenado hay tres alternativas muy adecuadas. La primera opción es utilizar un sensor de ultrasonidos, o un sensor óptico o un sensor láser; la segunda, emplear un sensor LIDAR (*Laser Imaging Detection and Raging*); y la tercera, implementar un sistema de análisis de imagen.

Los tres tipos de sensores mencionados para la primera solución se basan en el mismo concepto y también son conocidos como sensores TOF (*Time of Flight*). El sensor emite una onda y recibe la onda reflejada que retorna, cronometrando el tiempo entre la emisión y la recepción. Con el tiempo de ida y vuelta de la onda se puede calcular la distancia al objeto o elemento en el que rebota la onda según la siguiente fórmula:

$$\text{Distancia } D = \frac{1}{2} \cdot t \cdot C, \text{ siendo } C \text{ la velocidad del sonido o la luz según el tipo de sensor.}$$

La diferencia entre los tres tipos de sensores es el tipo de onda emitida. En el caso del sensor ultrasónico se emite una onda ultrasónica, en el sensor óptico un haz de luz infrarrojo y en el sensor láser un rayo láser. Las características físicas de cada tipo de onda determinan el funcionamiento del sensor.

PARÁMETRO	Ultrasónico	Óptico	Láser
Detección de objetos	Afectada por materiales que absorban el sonido.	Detección afectada por materiales/colores del objeto	Detección afectada por materiales/colores del objeto
Precisión	Baja	Alta	Muy Alta
Velocidad de respuesta	Lenta	Rápida	Rápida
Polvo/agua	Inmune	Afectado	Afectado

Tabla 11. Comparativa de las características generales de los distintos sensores TOF.



Imagen 10. Sensor ultrasónico HC-SR04, sensor óptico SHARP GP2Y0A02YK0F y sensor láser VL53L0X.

LIDAR es una tecnología que va un paso más allá respecto el sensor láser. Un sensor LIDAR emite haces de luz láser infrarroja que se reflejan en los objetos y, al retornar, son leídos por la lente receptora infrarroja del lector. De esta manera el procesador del LIDAR obtiene una nube de puntos de su entorno y para cada punto se conoce su posición precisa en el espacio y la distancia que hay hasta él. El campo de medición de los sensores LIDAR es típicamente de 360 grados y este sensor es empleado habitualmente en los coches autónomos y los robots de limpieza para monitorizar el entorno. El uso de esta tecnología se está extendiendo cada vez más y ya se han creado sensores compatibles con Arduino y Raspberry Pi para desarrollar proyectos de robótica.



Imagen 11. Sensores LIDAR: YDLIDAR X4 y RPLIDAR A2M8 respectivamente.

Por último, la tercera opción se basa en confeccionar un sistema de análisis de imagen mediante una cámara y un *software* de procesamiento de imágenes. En este caso sería necesario utilizar Raspberry Pi, dado que, como se ha comentado previamente, se requiere de una potencia de procesamiento superior a la de las placas Arduino.

5.4.4. Conectividad a Internet

Con el objetivo de que el contenedor pueda acceder a una base de datos o pueda participar en el Internet de las Cosas o IoT, el contenedor tiene que poder conectarse a Internet. En el caso de desarrollar el contenedor con Arduino, si se emplean los modelos más populares como el Arduino UNO o Arduino Mega, sería necesario añadirle un módulo o *shield* de conectividad. Únicamente algunos modelos de Arduino incorporan conectividad WiFi o Ethernet por defecto. En el caso de desarrollarlo con Raspberry Pi esto no sería necesario, dado que los componentes para conectarse a Internet ya están integrados en la placa.

Al igual que con los ordenadores, para conectar Arduino a Internet se dispone de dos tipos de conectividad que son la red WiFi y la red Ethernet. Ambas conectividades permiten acceder a la red con fiabilidad y seguridad, estableciendo una conexión veloz y estable. A grandes rasgos se puede asociar la palabra WiFi a la conexión a la red sin necesidad de cables, mientras para una conexión Ethernet es necesario un cable. Al ser una conexión cableada, Ethernet supera en velocidad y estabilidad a la conexión WiFi.

Como la conectividad a la red es muy importante actualmente y la mayoría de dispositivos disponen de ella, hay una gran cantidad de módulos y *shields* que capacitan al Arduino para conectarse a la red. En el caso del WiFi, la mayoría de los componentes se basan en uno de los tres siguientes chips: CC3000, ESP8266 y Ralink. Estos chips se han hecho populares debido a su reducido coste y pequeño tamaño, siendo el más utilizado el chip ESP8266. Por último, si se quiere disponer de una conexión más potente también hay una gran variedad de módulos y *shields* para incorporar la conectividad Ethernet.

Las características de los módulos o *shields* de conectividad, tanto WiFi como Ethernet, más populares para Arduino son las siguientes:

MODELO	WIFI				
	WiFi Shield	ESP-01	HLK-RM04	Shield W5100	ENC28J60
Chip base	CC3000	ESP8266	Ralink	W5100	ENC28J60
Voltaje de trabajo	4,5-12V	3,3V	5V	5V	3,3V
Interfaz	SPI	UART	UART	SPI	SPI

Tabla 12. Características generales de los componentes de conectividad WiFi.

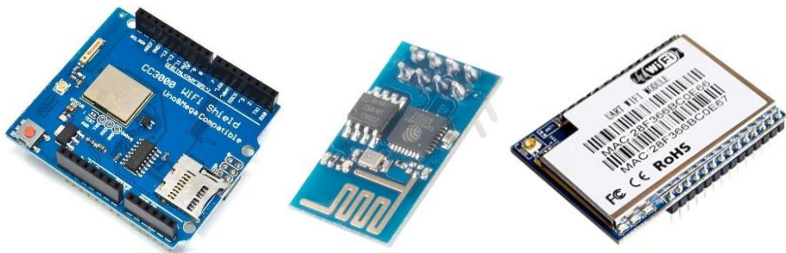


Imagen 12. WiFi Shield CC3000, módulo ESP-01 y módulo HLK-RM04, respectivamente.

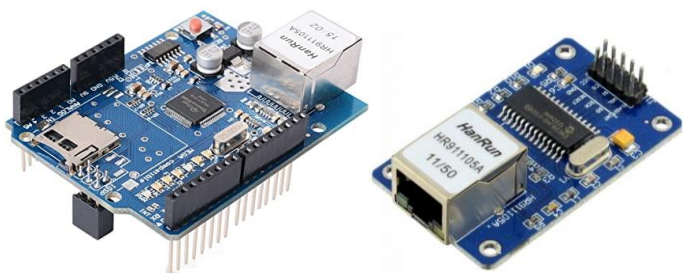


Imagen 13. Shield W5100 y módulo ENC28J60, respectivamente.

5.4.5. Visualización de la información

Por último, es preciso completar la estructura del dispositivo con un sistema que visualice la información extraída de las etiquetas y que se quiere transmitir al usuario o informar al usuario de los procesos que el dispositivo está ejecutando. Hay un gran abanico de posibilidades para desempeñar dicha función, habiéndose seleccionado las siguientes:

- Un conjunto de leds, uno para cada tipo de fracción. Al leer el producto se enciende el led correspondiente al contenedor al que debe ir depositado. Es una opción muy sencilla pero que puede mostrar información muy simple y limitada.
- Una pantalla de cristal líquido (en adelante, LCD, del inglés *Liquid-Crystal Display*), o un visualizador matricial o una pantalla OLED (*Organic Light-Emitting Diode*) que muestre en pantalla el contenedor correcto para la deposición del desecho.

Una pantalla LCD está formada por un número de píxeles en color colocados delante una fuente de luz o reflectora. Suelen ser delgadas y planas y su consumo eléctrico es bajo. Existen muchos módulos de pantalla LCD diseñados para trabajar con Arduino o Raspberry Pi y son muy comunes y utilizados en gran cantidad de proyectos de prototipado. El módulo se encarga de convertir las señales eléctricas de la placa en información visual y su implementación es muy sencilla. Se utiliza principalmente para mostrar información formada por caracteres alfanuméricos.

Un visualizador de matriz es un dispositivo para mostrar información mediante un conjunto de leds. Típicamente, los múltiples leds se distribuyen rectangularmente y existen varios tamaños, siendo el más popular los cuadrados 8x8. Aunque su resolución es limitada, el visualizador de matriz permite mostrar textos, dibujos y animaciones. Se pueden combinar varias matrices para conformar un visualizador de mayor tamaño. Al igual que las pantallas LCD, existen muchos módulos compatibles para operar con Arduino y Raspberry Pi. Dichos módulos son controladores diseñados específicamente para gestionar las matrices led ya que controlar una matriz directamente con la placa ocuparía una gran cantidad de pines.

Una pantalla OLED está formada por un conjunto de leds especiales en el que la capa emisiva está formada por un compuesto orgánico que emite luz en respuesta a la electricidad. Las pantallas OLED compatibles con Arduino y Raspberry Pi son menos populares que las otras dos opciones presentadas y tienden a ser de tamaño reducido y muy delgadas. Su montaje es sencillo pero su programación es compleja y ocupa gran cantidad de memoria. Las pantallas OLED tienen la ventaja de tener un consumo muy bajo, mejor visibilidad en ambientes luminosos y la capacidad de hacer

configuraciones de la pantalla muy complejas. Los módulos que operan con Arduino y Raspberry Pi incluyen el controlador y la electrónica necesaria para su correcto funcionamiento.



Imagen 14. Pantalla LCD, visualizador de matriz y pantalla OLED, respectivamente.

- Un sistema basado en un módulo de reconocimiento de voz que informe de la fracción del residuo. Debido al incremento y difusión de la interacción con ciertos dispositivos electrónicos por medio de la voz, la industria ha desarrollado módulos compatibles con Arduino y Raspberry Pi para simular dicha función. Es por ello, que existen varios módulos que permiten reconocer comandos y actuar en función. Sin embargo, el funcionamiento y prestaciones de los componentes disponibles están muy limitados, aún están lejos de emular a *Siri* o *Cortana*.

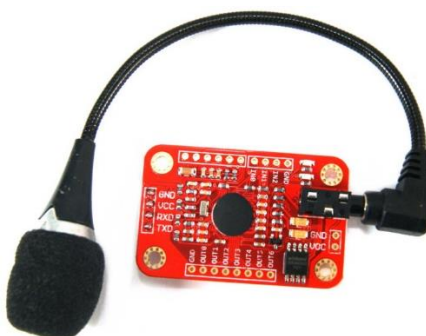


Imagen 15. Módulo de reconocimiento de voz Elechouse V3.

- Un sistema de avisos y notificaciones mediante el móvil. La información puede ser transmitida a éste sin necesidad de elementos físicos extra. La complejidad de este sistema recae en el *software* y se demandará mayor potencia al módulo de conectividad a la red.

6. Implementación del *hardware*

En este apartado se profundizará en la descripción de los componentes escogidos y se justificará su elección frente a las otras alternativas. Posteriormente, se describirá la arquitectura de los distintos sistemas, su proceso de implementación y los inconvenientes surgidos durante éste.

6.1. Alternativas tecnológicas escogidas

Para seleccionar qué componente emplear para cada función se han tenido en cuenta los aspectos técnicos de éste, su coste, la disponibilidad de documentación e información oficial y el soporte que recibe por parte de la comunidad *Maker*.

Hay que remarcar que el presupuesto del que se ha dispuesto es limitado y, por lo tanto, se ha optado por utilizar alternativas tecnológicas a disposición de la Universidad.

6.1.1. Placa de desarrollo: Arduino UNO

En primer lugar, se ha decidido desarrollar el dispositivo con Arduino. La justificación de esta decisión se basa en las ventajas de Arduino frente a Raspberry Pi que se han explicado en la sección anterior. Los puntos favorables de la plataforma Arduino son su versatilidad, por la compatibilidad con gran cantidad de distintos componentes, su sencillez y rapidez para implementar el *software* y su mejor comunicación con cualquier tipo de sensor.

En este proyecto se ha escogido la placa de desarrollo Arduino UNO, que incorpora el microcontrolador *Atmel ATmega328*. De entre las decenas de modelos diferentes de placas Arduino, se ha escogido el Arduino UNO por disponibilidad en la Universidad, porque se ha considerado que sus prestaciones y características son suficientes para operar el prototipo y sus periféricos y porque hay un mayor número de proyectos que emplean este modelo. Se enumeran las principales características del Arduino UNO en la tabla 13.

PARÁMETRO	VALOR
Microcontrolador	ATmega328
Voltaje operativo	5V
Voltaje de entrada recomendado	7-12V
Pines digitales de Entradas/Salidas	14, 6 de ellos salidas PWM
Pines analógicos de entrada	6
Puertos serie	1, conectado con la conexión USB
Memoria Flash	32kB
SRAM	2kB
EEPROM	1kB
Frecuencia de reloj	16MHz

Tabla 13. Características del Arduino UNO

Puede consultarse toda la información referente al Arduino UNO en («Arduino Uno Rev3 | Arduino Official Store», s. f.).



Imagen 16. Arduino UNO

Además, para conectar el Arduino UNO a los periféricos, se han utilizado complementos básicos como cables de conexión y una *protoboard*.

6.1.2. Sistema RFID

A continuación se describen los módulos lectores y etiquetas escogidos para su respectiva banda de frecuencia.

6.1.2.1. LF

Para su implementación se ha utilizado el lector RDM6300 y etiquetas EM4100. El módulo RDM6300 es una versión más económica del módulo RDM630 (Elti Eletronika, 2019), con la misma disposición de los pines y el mismo protocolo de transferencia. El lector RDM6300 se controla mediante comandos seriales TTL-UART, por lo que es compatible con cualquier microcontrolador Arduino. El módulo incorpora una antena y es compatible con los *tags* EM4100 de solo lectura o lectura/escritura.

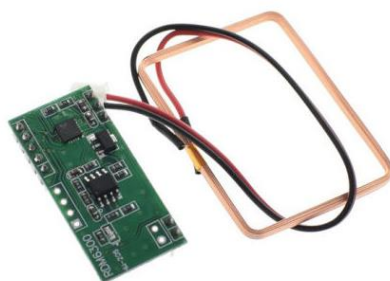


Imagen 17. Módulo RFID de baja frecuencia RDM6300

En la siguiente tabla se muestran sus principales características:

PARÁMETRO	VALOR
Frecuencia	125kHz
Velocidad de transmisión	9600
Interfaz	TTL, RS232
Voltaje de trabajo	5V
Corriente de consumo	<50mA
Distancia de lectura	20-50mm
Tamaño del PCB	38.5x19x9mm
Temperatura de operación	-10 a 70°C
Humedad de operación	0-95%

Tabla 14. Características del módulo RDM6300

A fin de que el lector sea capaz de leer la información guardada en una etiqueta RFID, éste debe saber cómo está guardada la información y el protocolo para extraerla. EM4100 es el protocolo más común para el formato de los datos de etiquetas de baja frecuencia. Las

etiquetas EM4100 son etiquetas pasivas con 64 bits de memoria de solo lectura, es decir, no se puede modificar o escribir información adicional después de la primera escritura inicial.



Imagen 18. Etiquetas EM4100 de 64 bits.

La distribución de los datos en las etiquetas es la siguiente («EM4100 protocol description», s. f.):

1	1	1	1	1	1	1	1	1	Encabezado de 9 bits, todos 1's siempre.
ID del cliente de 8 bits.	D00	D01	D02	D03	P0	Cada grupo de 4 bits va seguido con un bit de paridad par.			
	D04	D05	D06	D07	P1				
	D08	D09	D10	D11	P2				
	D12	D13	D14	D15	P3				
32 bits de Datos.	D16	D17	D18	D19	P4				
	D20	D21	D22	D23	P5				
	D24	D25	D26	D27	P6				
	D28	D29	D30	D31	P7				
	D32	D33	D34	D35	P8				
	D36	D37	D38	D39	P9				
	4 columnas bits de paridad	PC0	PC1	PC2	PC3		S0	Bit de parada (0).	

Figura 5. Configuración de los datos en los tags EM4100.

Al coger energía del campo electromagnético, el tag empieza a transmitir los datos con el formato mencionado. Los bits del inicio son una marca secuencial para determinar el inicio de la cadena de datos. Esta secuencia de nueve unos seguidos no se puede repetir gracias al uso de los bits de paridad par después de cada conjunto de 4 bits. La etiqueta RFID continúa transmitiendo esta cadena de datos repetidamente mientras tenga energía suficiente. Debajo se muestra un ejemplo de la cadena de datos que la etiqueta enviaría al lector repetidamente:

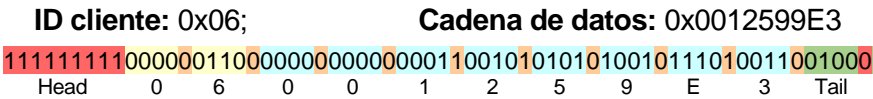


Figura 6. Ejemplo de cadena de datos que transmite una etiqueta EM4100 al estar dentro del rango del lector.

6.1.2.2. HF

El Sistema RFID de HF se ha implementado utilizando el módulo NFC/RFID de *elecFreaks* que se basa en el chip PN532, uno de los chips NFC más populares. El módulo NFC opera a una frecuencia de 13,56MHz, incorpora una antena integrada y se alimenta a 5V.



Imagen 19. Módulo NFC de elecFreaks.

La siguiente tabla presenta las principales características del lector:

PARÁMETRO	VALOR
Frecuencia	13,56MHz
Velocidad de transmisión	9600
Interfaz	SPI, I2C
Voltaje de trabajo	5V
Corriente de consumo	<90mA
Distancia de lectura	20-50mm
Tamaño de la antena	46x32x3mm
Tamaño del PCB	38.5x19x9mm
Temperatura de operación	-30 a 85°C
Humedad de operación	0-95%

Tabla 15. Características generales del módulo NFC de elecFreaks.

Se ha escogido utilizar etiquetas RFID MIFARE *Classic* 1K, que cumplen con la ISO1443 A. Son *tags* pasivos de lectura/escritura con una memoria EEPROM de 1024 bytes. Cuentan con códigos de seguridad para asegurar la protección de la información almacenada. El espacio disponible para el usuario tras descontar el espacio consumido por los códigos de seguridad es de 768 bytes.

Para poder hacer un correcto uso de las etiquetas es necesario conocer el formato de los datos contenidos. Los datos están divididos en 16 sectores de 4 bloques cada uno, 64 bloques en total. Cada bloque está formado por 16 bytes. A continuación, se describirá el formato y la

función de sectores y bloques preconfigurados («TUTORIAL-TAGS-ISO-14443A-PUNTO-FLOTANTE-S.A.», s. f.):

- **Número de serie o NUID (*Non Unique Identification Number*):** número de serie del *tag* formado por 4 bytes. Este identificador es pregrabado en fábrica. Son los primeros 4 bytes del Bloque 0 del Sector 0. Todo el Bloque 0 está protegido y solo puede leerse. Los bytes restantes del Bloque 0 son los datos del fabricante.

Bloque 0 / Sector 0																
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	NUID				Datos del fabricante											

Tabla 16. Configuración del Bloque 0, NUID y datos del fabricante.

- **Sector Trailer:** al cuarto bloque de cada sector se le llama “*Sector Trailer*”. Es el bloque encargado de los códigos de acceso para la lectura/escritura de los datos de su respectivo sector. Está compuesto por 2 códigos de seguridad, conocidos como Key A y Key B, y un conjunto de 4 bytes, llamados “*Access bits*”. Los tres conjuntos de bytes característicos vienen de fábrica con unos valores por defecto.
 - **Key A:** puede ser modificado, pero no leído. Su código por defecto es 0xFFFFFFFFFFFF. Al no poderse leer, los valores se muestran como 0's.
 - **Key B:** código de seguridad opcional, puede ser modificado y leído. Su código por defecto es 0xFFFFFFFFFFFF.
 - **Access bits:** definen los permisos para cada uno de los bloques del sector. Su código por defecto es 0xFF 0x07 0x80 0x69.

Sector Trailer (cuarto bloque de cada sector)																
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Key A						Access Bits				Key B					

Tabla 17. Configuración del sector Trailer, Key A, Key B y Access bits.

Para poder leer o modificar los datos de alguno de los sectores se requiere conocer la clave Key A y/o Key B. La configuración de los Access bits determinará el comando que se ejecutará al bloque. Hay 6 comandos aplicables: lectura, escritura, incremento, decremento, restaurar y transferir. Los valores de fábrica iniciales configuran la etiqueta RFID para permitir la lectura/escritura de datos en todos los bloques disponibles para guardar información.

En la imagen 20 podemos ver una parte (del bloque 0 hasta el bloque 25) de la información transmitida al monitor serie del IDE Arduino al ejecutar un programa que realiza la lectura completa de una etiqueta RFID. Se indican los elementos previamente mencionados en el mismo color.

7	6D 42 5A 26	53 8	4	0	62 63	64 65	66 67	68 69	Block 0	Manufacturer Block
8	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 1	Data Block
9	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 2	Data Block
10	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 3	Sector Trailer
11	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 4	Data Block
12	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 5	Data Block
13	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 6	Data Block
14	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 7	Sector Trailer
15	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 8	Data Block
16	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 9	Data Block
17	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 10	Data Block
18	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 11	Sector Trailer
19	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 12	Data Block
20	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 13	Data Block
21	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 14	Data Block
22	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 15	Sector Trailer
23	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 16	Data Block
24	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 17	Data Block
25	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 18	Data Block
26	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 19	Sector Trailer
27	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 20	Data Block
28	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 21	Data Block
29	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 22	Data Block
30	0 0 0 0	0 0	FF 7	80 69	FF FF	FF FF	FF FF	FF FF	Block 23	Sector Trailer
31	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 24	Data Block
32	0 0 0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	Block 25	Data Block

Imagen 20. Formato de los datos de un tag MIFARE (ISO14443 A).

6.1.2.3. UHF

Según la justificación en la sección anterior, en teoría, tanto la banda de LF como las de HF y UHF son adecuadas para el desarrollo del dispositivo. En la práctica, sin embargo, está última actualmente aún está lejos de ser viable económicamente. Es cierto que en el mercado están disponibles algunos componentes, aunque muy pocos y con un alto precio, todos alrededor de los 200€. Además, al ser componentes caros, su uso en la comunidad *Maker* no es muy popular, por lo que los módulos no cuentan con mucho soporte, documentación o librerías propias.

El objetivo de un prototipo es prever cómo funcionaría aproximadamente un dispositivo final reduciendo la inversión necesaria. Por lo tanto, dada la gran inversión de dinero y tiempo, y que se pueden valorar los posibles beneficios frente a un sistema HF, se ha decidido descartar la implementación de un sistema UHF en la práctica.

Con todo, igualmente, se expondrá la solución que se habría escogido en el caso de haber desarrollado el prototipo con el sistema RFID de UHF, la justificación de su elección y cómo habría sido la supuesta estructura del sistema y ventajas.

Para desarrollar el sistema RFID UHF se habría escogido el lector *Sparkfun M6E-NANO*. Para decantarse por una opción es muy importante saber de antemano cómo trabaja el lector, qué limitaciones tiene y los requisitos para hacerlo operable. En este caso, aunque su coste es superior al lector *Cottonwood*, el lector de *Sparkfun* cuenta con información oficial muy detallada y mayor cantidad de documentación. Además, la página web de *Sparkfun* dispone de una guía que explica cómo realizar la instalación del lector, el *hardware* requerido y las características y prestaciones una vez puesto en marcha («Simultaneous RFID Tag Reader

Hookup Guide», s. f.). El lector UHF *Sparkfun M6E-NANO* es compatible con etiquetas EPC gen2 y puede leer, simultáneamente, hasta 150 *tags* por segundo. Tiene una antena integrada con un rango de lectura de 60cm, pero también permite la conexión de una antena externa. Con la antena adecuada se pueden conseguir distancias de lectura de hasta 5m. En la tabla 18 se muestran sus características principales.

PARÁMETRO	VALOR
Frecuencia de trabajo	840 - 960MHz
Velocidad de transmisión	9600, 115200
Interfaz	FTDI, UART
Etiquetas compatibles	ISO18000-6C (EPC gen2)
Voltaje de trabajo	5V
Corriente de consumo	1A
Distancia de lectura/escritura	Hasta 5m
Potencia de lectura/escritura	0dBm hasta 27dBm
Tiempo medio de lectura	80ms (96-bit EPC)
Tamaño del PCB	78x53mm

Tabla 18. Características principales del lector UHF *Sparkfun M6E-NANO*.

En el desarrollo del dispositivo con el sistema RFID de UHF se habrían utilizado etiquetas EPC Class1 Gen2. Debido a que no se realizará la implementación del sistema UHF, se introducirán brevemente las principales características de este modelo de etiquetas.

El estándar EPC *Class 1 Gen 2* fue diseñado para optimizar la lectura simultánea de *tags* mediante un sistema de anticollisión. Son etiquetas de alta seguridad y se fabrican con diferentes tamaños de memoria según las necesidades del cliente. Cada *tag* cuenta con dos claves de seguridad llamadas “*Kill password*” y “*Access password*”.

- ***Kill password***: contraseña de 32 bits que permite silenciar la etiqueta. Su valor por defecto es 0 y el comando “*Kill*” solo se ejecuta si la contraseña ha sido establecida (es diferente de 0).
- ***Access password***: contraseña de 32 bits que establece el *tag* en modo seguro permitiendo realizar operaciones privilegiadas. Su valor por defecto es 0.

EPC *Class 1 Gen2* es el estándar adoptado por la mayoría de fabricantes de sistemas RFID de UHF, siendo el principal protocolo actualmente.

6.1.3. Sensor de ultrasonido: HC-SR05

En la sección anterior se plantearon, entre otras, dos funcionalidades que podría agregar el contenedor: contabilizar la cantidad de residuos generados o monitorizar el nivel de llenado del contenedor. De estas dos opciones se descarta la primera, dado que la segunda función se considera más útil y responde a las necesidades del usuario. Además, al implementar el sistema con células de carga surgen varios inconvenientes. La primera solución obvia sería instalar una célula de carga en la parte inferior del tamaño de la base del contenedor. Al depositar un residuo, el usuario no invierte tiempo en colocar el residuo en una posición óptima u organizada, lo que provoca que finalmente la distribución de los residuos sea aleatoria y no uniforme. Dicha distribución puede ocasionar que el peso de los residuos se distribuya por las paredes del contenedor, disminuyendo la fuerza que aplican a la célula de carga y, por tanto, dando un resultado erróneo de la masa de residuos generada. En segundo lugar, el tamaño habitual de las células de carga en el mercado es muy inferior al tamaño de la base de un contenedor cotidiano. Si se opta por utilizar las células de carga que se combinan con una plataforma y un amplificador, el precio se incrementa por encima de las demás alternativas. Por ambos motivos se ha priorizado monitorizar el nivel de llenado respecto a contabilizar la cantidad de residuos reciclados, convirtiéndose esta contabilización en una posible mejora para líneas futuras.

De las tres opciones presentadas para monitorizar el nivel de llenado del contenedor se ha decidido proceder con la primera. Los sensores LIDAR tienen un alto coste y son más apropiados para dispositivos que se mueven. Se estarían desperdiciando sus altas prestaciones y su alto coste no justifica su inclusión en el contenedor. El sistema de análisis de imágenes se descarta al haber elegido la plataforma Arduino, dado que sería necesaria una gran potencia de procesamiento para ejecutar su *software*. Además, al igual que los sensores LIDAR, las prestaciones, costes y tiempo de inversión en un sistema de análisis de imagen sobrepasan el objetivo de la función.

Por ello ambas soluciones son descartadas y se decide proceder con la implementación de un sensor de ultrasonidos. Las ventajas con las que contribuye este transductor frente al sensor óptico o al sensor láser, es que es más económico y que la detección no se ve afectada por la humedad, ni la suciedad ni materiales transparentes o brillantes. Sin embargo, los otros dos tipos de sensores, generalmente, lo superan en precisión y rango de medición y además son más estables ante las interferencias causadas al utilizarse en combinación con más sensores del mismo tipo. Aun así, se ha considerado que estos puntos débiles del sensor ultrasónico no son relevantes para este proyecto, dado que la precisión y el rango de medición son más que suficientes y que se va a emplear únicamente un sensor.

El sensor de ultrasonidos escogido es el módulo HC-SR05, una versión actualizada del sensor HC-SR04. Este último sensor es el sensor ultrasónico más económico, versátil y utilizado de

todo el mercado. Los cambios entre las dos modelos son una ligera mejora de la precisión (2mm frente a 3mm) y HC-SR05 dispone de un pin llamado OUT que permite dos modos de funcionamiento distintos.



Imagen 21. Módulo sensor de ultrasonido HC-SR05.

A continuación, se muestran sus características principales:

PARÁMETRO	VALOR
Voltaje de trabajo	4,5 hasta 5 V
Frecuencia de la onda	40 kHz
Resolución de la medición	2mm
Ángulo de medición	15°
Corriente	10 a 40 mA
Distancia de detección	2 hasta 450 cm

Tabla 19. Características generales del módulo HC-SR05.

6.1.4. Módulo de conectividad a Internet: ESP-01

Se ha decidido dotar al Arduino de conectividad WiFi debido a que permite acceder a la red de manera inalámbrica y es mucho más versátil. Como no se realizarán operaciones o transmisiones muy grandes, la estabilidad y velocidad que ofrece la conectividad WiFi es suficiente.

De las tres opciones WiFi presentadas se ha descartado el *shield* Wifi, basado en el chip CC3000, debido a que tiene un precio superior. En cambio, se descarta el módulo HLK-RM04 porque es más complejo de implementar y no hay una documentación muy detallada por parte del proveedor.

Por lo tanto, el módulo que se va a encargar de dotar de conectividad WiFi al Arduino es el ESP-01, basado en el chip ESP8266. El chip ESP8266, diseñado inicialmente con la IoT en mente, se popularizó por ser el primer dispositivo realmente económico que proporcionaba conectividad WiFi y por su pequeño tamaño. En realidad, ESP8266 es un microprocesador completo con WiFi integrado controlado mediante comandos AT. Lo que significa que

prácticamente es capaz de actuar igual que las placas Arduino. Su potencia es algo inferior a la del Arduino UNO y tiene ciertas limitaciones que se comentarán más adelante. Sin embargo, en ciertos proyectos con pocos periféricos, el ESP8266 puede actuar como el procesador principal evitando tener que utilizar las placas Arduino. Para ello, es posible reprogramar su *firmware* y volcarle programas. Además, este chip se utiliza tanto por la comunidad *Maker* que una parte de esta ha desarrollado un *plugin* para el IDE de Arduino. Es decir, se puede programar el ESP8266 desde el IDE de Arduino, utilizando los mismos comandos y lenguaje, e incluye una serie de librerías que simplifican su configuración y permiten operar con mayor versatilidad.

En conclusión, hay la posibilidad de incorporar el ESP-01 como un módulo adicional a Arduino para dotarlo de conectividad, mediante un puerto serial, o se puede optar por utilizarlo como procesador principal. El ESP8266 se alimenta con 3,3V y cuenta con dos pines de propósito general (en adelante, GPIO, del inglés *General Purpose Input/Output*), en la tabla 12 se muestran sus características generales.

6.1.5. Pantalla LCD

La opción de mostrar la información mediante un sistema de leds se ha descartado porque es muy sencilla y no permite visualizar información compleja eficientemente. A mayor complejidad de la información que se quiere mostrar mayor número de leds necesarios, y por lo tanto, número de pines utilizados. Debido a que el Arduino UNO dispone de un número limitado de pines que deben emplearse para el control de los demás periféricos, se descarta el sistema de leds.

Al contrario, el sistema de reconocimiento de voz se descarta por ser complejo y sus prestaciones no son necesarias para la información que se quiere exponer. También se descarta el sistema de avisos por móvil por su complejidad, horas de inversión necesarias en su *software* y porque no se quiere sobrecargar al módulo ESP-01. Este sistema se mantiene como posible línea futura.

Por último, de la opción restante se descarta la pantalla OLED porque requiere gran cantidad de memoria y se desconoce cuánta hay disponible para el desarrollo del sistema de visualización de información. Sus ventajas frente las demás pantallas no justifican su inclusión. Finalmente, entre la matriz de leds y la pantalla LCD se escoge implementar la pantalla LCD por disponibilidad en la Universidad y debido que ambas tienen prestaciones muy similares.

6.2. Arquitectura final del prototipo

En las siguientes imágenes se puede observar el montaje del contenedor doméstico inteligente. Para cambiar de un sistema RFID al otro solamente es necesario conectar los cables de alimentación (*Vcc* y *GND*) del sistema RFID deseado y volcar su programa correspondiente al Arduino.



Imagen 22. Cubo con el sensor HC-SR05 incorporado.



Imagen 23. Montaje completo del contenedor doméstico.

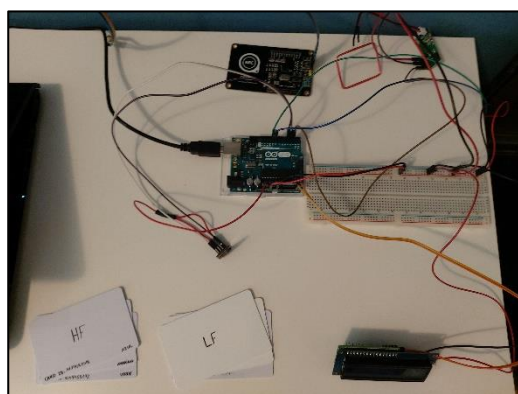


Imagen 24. Montaje del cableado general.

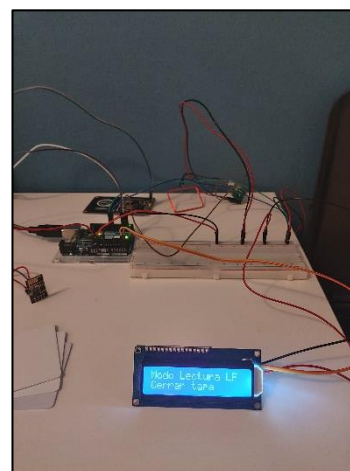


Imagen 25. Cableado del dispositivo en funcionamiento.

6.2.1. Sistema RFID de LF

En la siguiente imagen y tablas se especifica el tipo de conexión y la conexión de los pines de cada componente con el Arduino:

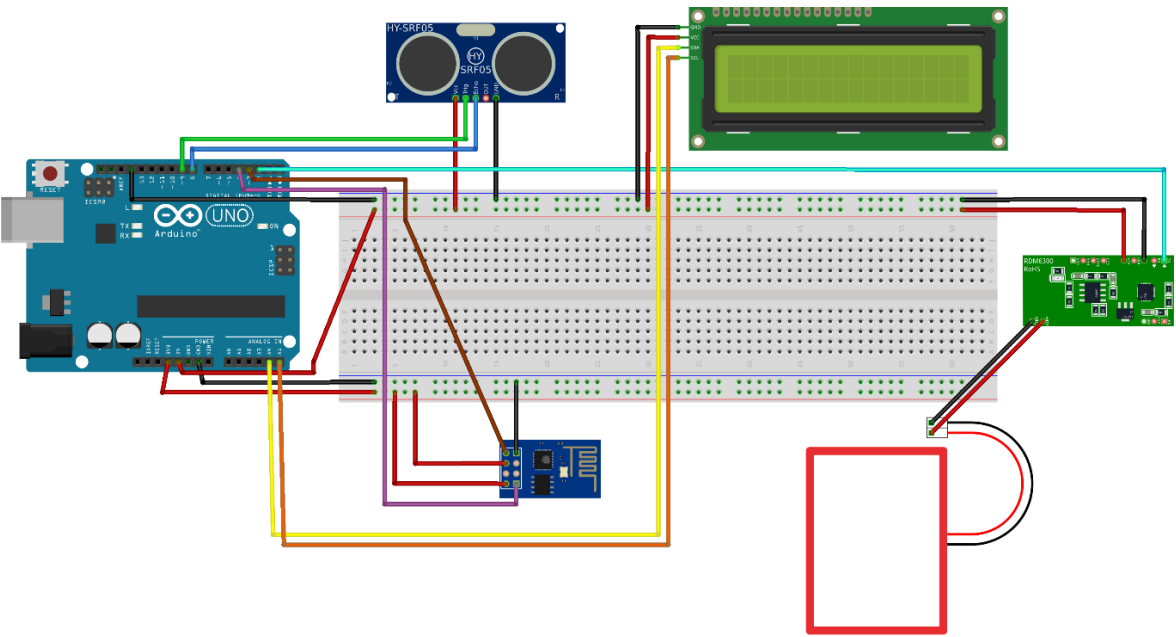


Imagen 26. Esquema de montaje del dispositivo con el sistema RFID de LF.

Tipo de conexión	ESP-01	Arduino UNO
UART (puerto serial)	Vcc	3,3 V
	GND	GND
	TX	D3
	RX	D4
	CH_PD	3,3 V

Tipo de conexión	RDM6300	Arduino UNO
UART (puerto serial)	Vcc	5 V
	GND	GND
	TX	D2

Tipo de conexión	HC-SR05	Arduino UNO
Interfaz mediante pulso de eco	Vcc	5 V
	GND	GND
	Echo	D8
	Trig	D9

Tipo de conexión	LCD	Arduino UNO
I2C	Vcc	5 V
	GND	GND
	SDA	A4
	SCL	A5

Figura 7. Tablas del cableado del dispositivo con sistema RFID de LF.

Durante la implementación del sistema RFID de LF, han surgido dos inconvenientes en relación con las características y funcionamiento del *hardware*. Ambos inconvenientes han sido solucionados mediante el *software*:

- **Lecturas múltiples:** RDM6300 es un lector muy económico y sencillo, por lo que incorpora un *firmware* de lectura de etiquetas limitado. Al acercar un tag, éste envía su información repetidamente hasta no disponer de suficiente energía. El RDM6300 trata esta información, pero no filtra en función de si la misma etiqueta ha sido leída numerosas veces en un corto período de tiempo. El problema surge cuando se trata

de ejecutar un programa después de cada lectura. Al recibir la misma lectura en un intervalo de tiempo muy corto dicho programa se ejecuta múltiples veces.

- **Recepción simultánea de datos de 2 puertos seriales:** el Arduino UNO cuenta con un puerto *hardware* serie (pines 0 y 1) conectado físicamente al conector USB para facilitar la conexión entre el ordenador y la placa. Si el ordenador y el Arduino se comunican mediante el monitor serie, estos pines digitales no se pueden utilizar. Cuando se necesitan más puertos serie es necesario trabajar con la librería “*SoftwareSerial*” que replica las funcionalidades de la comunicación serie. Sin embargo, tiene sus limitaciones. Una de ellas es que el Arduino no puede leer la información transmitida por más de un puerto serie. En el sistema RFID de LF son necesarios dos puertos serie para la comunicación entre el Arduino-ESP-01 y Arduino-RDM6300.

En el capítulo 7. Implementación del *software* se explican las soluciones aplicadas para resolver ambos problemas.

6.2.2. Sistema RFID de HF

En la siguiente imagen y tablas se especifica el tipo de conexión y la conexión de los pines de cada componente con el Arduino:

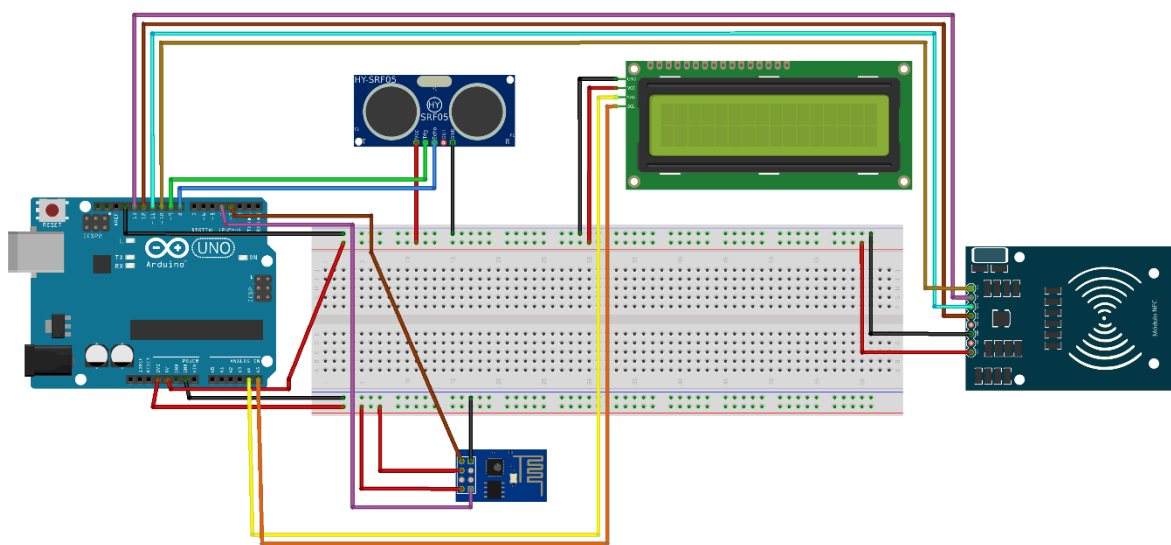


Imagen 27. Esquema de montaje del dispositivo con sistema RFID de HF.

Tipo de conexión	ESP-01	Arduino UNO
UART (puerto serial)	Vcc	3,3 V
	GND	GND
	TX	D3
	RX	D4
	CH_PD	3,3 V

Tipo de conexión	Módulo NFC	Arduino UNO
SPI	Vcc	5 V
	GND	GND
	SCK	D13
	MISO	D12
	MOSI	D11
	SCL/SS	D10

Tipo de conexión	HC-SR05	Arduino UNO
Interfaz mediante pulso de eco	Vcc	5 V
	GND	GND
	Echo	D8
	Trig	D9

Tipo de conexión	LCD	Arduino UNO
I2C	Vcc	5 V
	GND	GND
	SDA	A4
	SCL	A5

Figura 8. Tablas del cableado del dispositivo con sistema RFID de LF.

Durante la implementación del dispositivo con sistema RFID de HF no han surgido problemas debido a la compatibilidad de las distintas conexiones entre los periféricos y el Arduino. Además, el *firmware* y *software* tras el lector es muy sólido y hay librerías desarrolladas por la comunidad que facilitan su programación.

7. Implementación del *software*

En este apartado se introducirá y describirá el *software* que hace posible el funcionamiento del dispositivo y se documentarán las pruebas realizadas para determinar las prestaciones de cada sistema. El código completo de todos los programas *software* se puede ver en los Anexos II, III y IV.

7.1. Funciones generales

Se van a comentar las funciones generales incluidas en ambos sistemas RFID y que se encargan del funcionamiento de los componentes implementados en común.

7.1.1. Función *ultrasonicSensor()*

La función *ultrasonicSensor()* se encarga de calcular el nivel de llenado del contenedor. Es llamada al inicio de la función *loop*, por lo que se ejecuta reiteradamente.

Cuando es llamada, cambia el estado del pin TRIG del sensor ultrasónico a *HIGH*, enviando un pulso de onda. Al cabo de 10 microsegundos se vuelve a establecer el pin TRIG al estado *LOW*, deteniendo el envío de pulso de onda. La función *pulseIn* obtiene el tiempo que tarda el pin ECHO en cambiar entre estados. Este tiempo coincide con el tiempo entre que se ha enviado la onda y ha sido recibida de vuelta por el sensor ultrasónico. Con el tiempo obtenido se realizan varios cálculos para finalmente obtener el nivel de llenado del contenedor. Por último, envía la orden de mostrar el porcentaje de llenado en la pantalla LCD.

```
void ultrasonicSensor() {  
  
    digitalWrite(pin_TRIG, HIGH); //Envia el pulso para iniciar el sensor  
    delayMicroseconds(10);  
    digitalWrite(pin_TRIG, LOW);  
    tiempoSenal = pulseIn(pin_ECHO, HIGH); //Tiempo que dura el pin ECHO en HIGH  
    distanciaCM = (0.0343 * tiempoSenal) / 2; //Convierte el tiempo en distancia  
    (cm)  
    float porcentaje = (32-distanciaCM)*100/32;  
    int parte_entera= int(porcentaje);  
    if (parte_entera<0){  
        printlcd("Cerrar tapa      ",1);  
        delay(500);  
    }  
    else {  
        String string_lcd = "Distancia: "+String(parte_entera)+"%  ";  
        printlcd(string_lcd,1); //Muestra en la pantalla la distancia en  
        centímetros  
        delay(500);  
    }  
}
```

Figura 9. Código de la función *ultrasonicSensor()*.

7.1.2. Función *conectWifi()*

La función *conectWifi()* se encarga de establecer la conexión WiFi mediante comandos AT. Esta función es ejecutada cuando el Arduino recibe el comando “Configurar Wifi” a través del monitor serie y se ha implementado incorporando la librería *ITEADLIB Arduino WeeESP8266* («GitHub - itead/ITEADLIB_Arduino_WeeESP8266», s. f.).

Cuando es llamada, a través del monitor serie, pide al usuario por el nombre de la red WiFi a la que se quiere conectar y la contraseña de ésta. Al introducir ambos parámetros en el monitor serie, son enviados desde el Arduino al ESP-01. Si se han introducido correctamente el nombre y la contraseña, se conectará a la red.

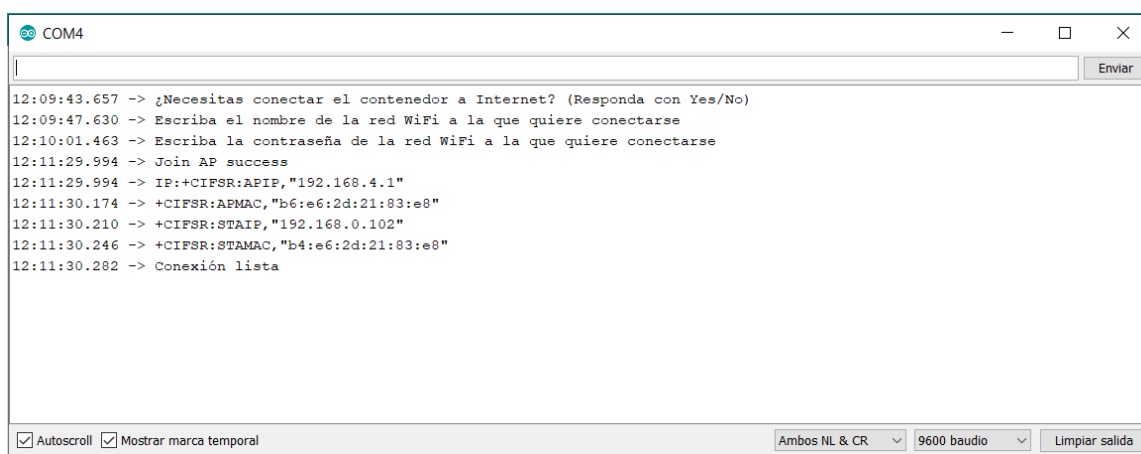


Imagen 28. Monitor serie del IDE de Arduino tras haber enviado el comando "Configurar Wifi" y haber introducido el nombre y la contraseña del Wifi correctamente.

7.1.3. Función *printLCD()*

La función *printLCD()* es una función sencilla para mostrar un *string* en la pantalla LCD. Es llamado cada vez que se quiere modificar el texto en pantalla o actualizarlo.

```

void printLCD(String txt, const int linea){
    int i=0;
    int longtxt=txt.length();
    while( i < longtxt) { //Mientras no haya timeout
        LCD.setCursor(i, linea);
        LCD.print(txt.charAt(i));
        i++;
    }
}
  
```

Figura 10. Código de la función *printLCD()*.

7.2. Software sistema RFID de LF

Para la puesta en marcha del dispositivo utilizando el sistema RFID de LF solo es necesario volcar al Arduino el programa principal *LecturaLF*. Dado que las etiquetas utilizadas no se pueden reescribir y el identificador viene asignado de fábrica, no es necesario el desarrollo de un programa de escritura de etiquetas. *LecturaLF* es el programa principal que integra las funciones generales descritas y desempeña la función de leer los tags. Se han definido dos estados del dispositivo para determinar cuál es la información que se quiere leer de entre el módulo ESP-01 y el lector RDM6300, debido al problema mencionado en 6.2.1:

- **Modo lectura LF:** estado principal, habilita la lectura de datos del lector RDM6300 y se actualiza el nivel de llenado del contenedor. Al ejecutarse el programa, el dispositivo se configura en este estado automáticamente.
- **Modo configurando WiFi:** estado secundario. El dispositivo entra en este estado al enviar la orden “Configurar Wifi” al Arduino a través del monitor serie. Mientras el dispositivo se encuentra en este estado, la lectura de *tags* está deshabilitada. Al cambiar a este estado, se ejecuta la función *connectWifi()* y, al finalizar, se vuelve al modo lectura LF.

Al utilizarse *tags* EM4100, que únicamente contienen el identificador, se requiere que éste se contraste en una base de datos que contenga el resto de la información del producto. Actualmente, esta base de datos está incluida dentro del programa *LecturaLF* para más sencillez. Sin embargo, modificando el programa se podría contrastar el identificador con una base de datos que estuviera en la red. Es importante explicar el protocolo de comunicación propio entre el lector RDM6300 y el Arduino, presente en la ficha técnica de este (Eltly Eletronika, 2019) y que no se debe confundir con el protocolo de comunicación entre las etiquetas EM4100 y el lector, explicado en 6.1.2.1. La etiqueta envía la cadena de datos descrita en dicho apartado que es recibida por el lector. A continuación el lector trata los datos recibidos para finalmente transmitir una cadena de datos como la siguiente:

Formato de los datos de salida (Interfaz TTL-RS232)

02	10 caracteres ASCII	Suma de verificación	03
Marca el inicio de la lectura.	Información que contiene el identificador del tag. Cada carácter ASCII se muestra como 2 valores Hexadecimales.	Comprobar que la secuencia de datos recibida es correcta.	Marca el final de la lectura.

Figura 11. Formato de los datos de salida del lector RDM6300.

Con el objetivo de leer correctamente el identificador de una etiqueta es necesario diseñar el programa de lectura teniendo en cuenta las marcas de inicio y fin de la secuencia de datos

que el Arduino recibe del lector. Puede ocurrir en muchas ocasiones que la información se envíe a medias o que ocurran alteraciones durante la lectura. Por ello, para mayor fiabilidad de la lectura se ha de controlar mediante ambas marcas. Además, debido al problema de múltiples lecturas descrito en 6.2.1 se ha almacenado la información leída de la etiqueta RFID previa. Finalmente, si la lectura se ha realizado correctamente y es la primera vez que la etiqueta es leída desde que se ha introducido en el campo de detección del lector, se compara el identificador con la base de datos. Si el identificador es conocido, la información del residuo relacionado es mostrada en el monitor serie del IDE de Arduino y el color de la fracción se muestra en la pantalla LCD.

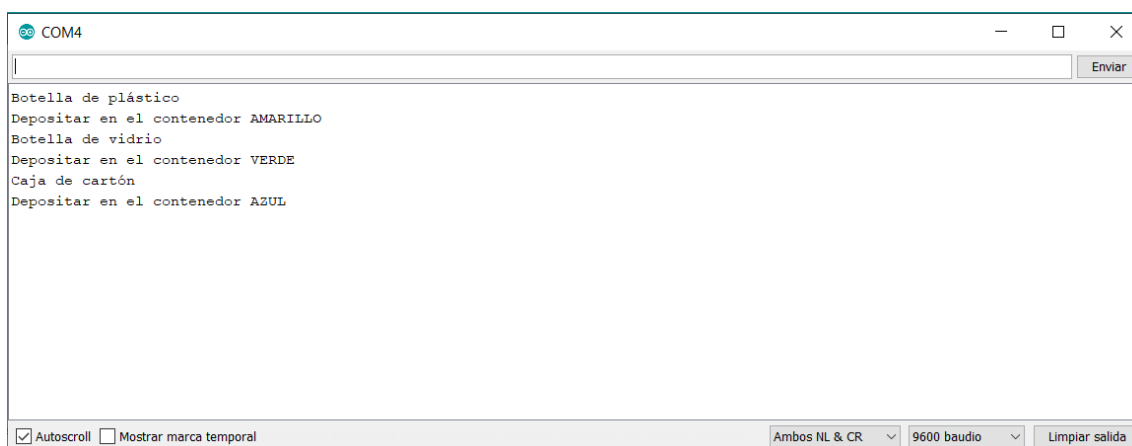


Imagen 29. Monitor serie del IDE de Arduino tras haber leído tres etiquetas RFID de LF distintas.

7.3. Software sistema RFID de HF

7.3.1. Software de grabación de etiquetas

El primer paso para poder realizar el estudio del funcionamiento del prototipo con el sistema RFID de HF, es introducir información en los *tags* RFID. Con dicho objetivo, se escribe un programa, llamado *EscrituraHF*, en el que se solicita el color de la fracción a la que debe ir depositado el producto. A continuación, el programa solicitará que se acerque la etiqueta al módulo PN532, para grabar en ella la información establecida. El color de la fracción se almacenará en el bloque 62 de las etiquetas MIFARE.

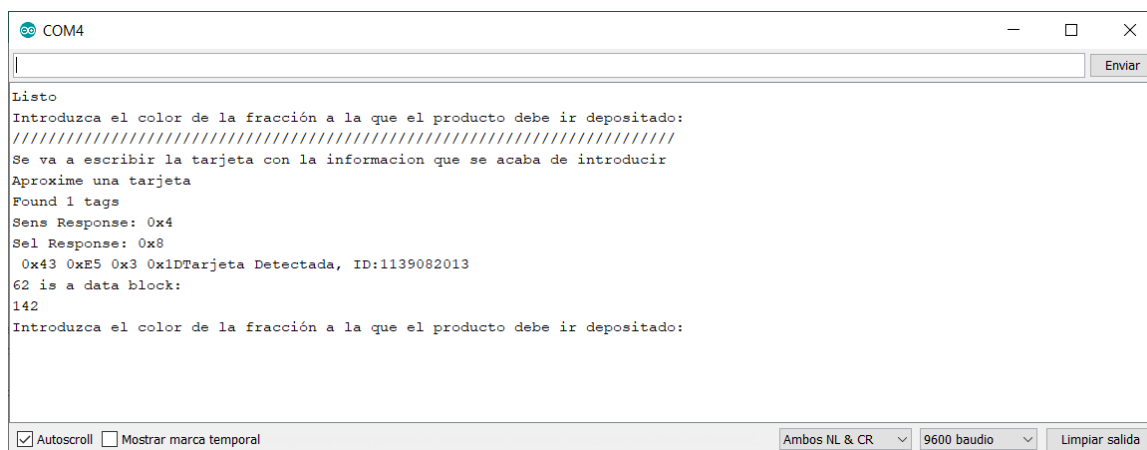


Imagen 30. Monitor serie del IDE de Arduino tras haber escrito una etiqueta RFID de HF.

Este programa es secundario: es necesario para realizar el proyecto, pero el usuario no tendrá acceso a esta funcionalidad. Simula y representa el proceso que debería llevar a cabo el fabricante al implementar los *tags* RFID en los consumibles.

7.3.2. Software de lectura de etiquetas

Se ha detallado cómo se graba información en las etiquetas RFID. Se procede entonces con la funcionalidad principal del dispositivo, la lectura de la información contenida en los tags. El *software* encargado de tal tarea es *LecturaHF*, que permite acceder y mostrar la información almacenada previamente en la etiqueta e, al igual que el archivo *LecturaLF*, incorpora las funciones generales descritas. Se ha implementado la lectura incorporando la librería *PN532*, que facilita la programación para la lectura de etiquetas MIFARE. La extracción de datos se desarrolla como se ha explicado en el apartado 6.1.2.2 en referencia a los *tags* MIFARE. Sencillamente, al acercar una etiqueta a una cierta distancia del lector se muestra la información contenida en el bloque 62 en el monitor serie. Tras la extracción de datos directamente se envía la orden de mostrar el color de la fracción en la pantalla LCD.

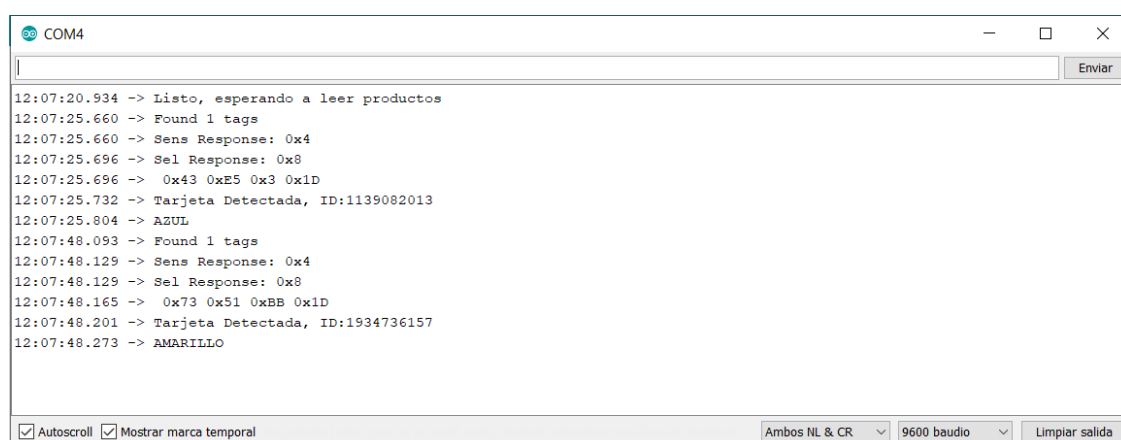


Imagen 31. Monitor serie del IDE de Arduino tras haber leído dos tags RFID de HF.

7.4. Pruebas y simulaciones

En este capítulo se documentarán las pruebas realizadas para determinar las prestaciones de los sistemas RFID de LF y HF. Las pruebas de rango y tiempo de lectura se han realizado para los dos tipos de sistema y la prueba de medición de nivel de llenado se ha realizado una única vez debido a qué es común para ambos sistemas.

7.4.1. Prueba de rango de lectura

Para realizar la prueba de rango se plantean tres escenarios para hacer las mediciones:

1. Etiqueta incorporada en residuo de “carácter neutral”.
2. Etiqueta incorporada en residuo que contiene líquido.
3. Etiqueta incorporada en residuo metálico.



Imagen 32. Residuos utilizados para las pruebas de rango y tiempo de lectura con etiquetas RFID incorporadas.

Se realizan 5 medidas para cada escenario, tanto para LF como HF, resultando en 30 mediciones, 15 y 15 respectivamente.

Escenario	DISTANCIA DE LECTURA (cm)					
	Baja frecuencia			Alta frecuencia		
	1	2	3	1	2	3
Medida 1	1,25	0,65	0,80	3,00	0,75	1,00
Medida 2	1,35	0,65	0,90	3,00	0,85	0,90
Medida 3	1,25	0,70	0,95	3,05	0,75	1,00
Medida 4	1,20	0,60	0,75	2,95	0,80	0,95
Medida 5	1,30	0,65	0,80	3,05	0,90	0,95
Media	1,27	0,65	0,84	3,01	0,81	0,96

Tabla 20. Resultados de la prueba de rango de lectura.

Distancia de lectura de los sistemas RFID de LF y HF

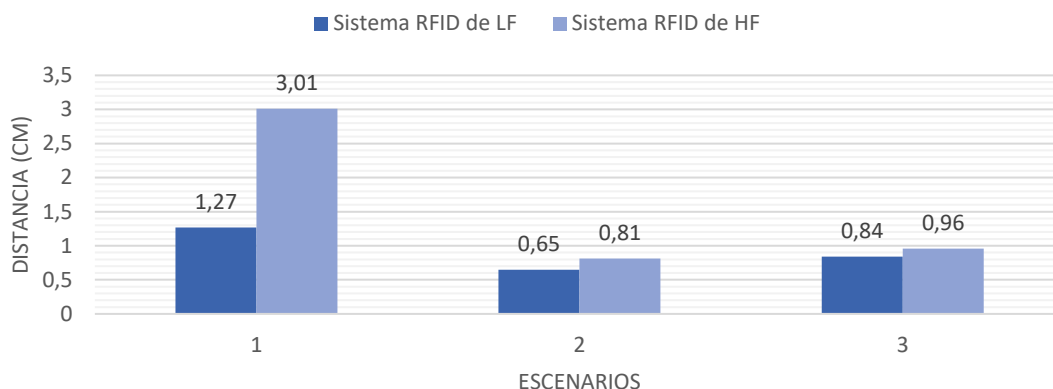


Gráfico 1. Resultados de la prueba de rango de lectura.

Gracias a los resultados de las pruebas se puede observar claramente como el material del residuo al que el *tag* va adherido afecta al rango de lectura/escritura. La distancia de lectura se ve afectada tanto por líquidos como materiales metálicos, siendo un tercio de la distancia de un residuo que no interfiere en la lectura/escritura.

7.4.2. Prueba de tiempo de lectura

Con los mismos escenarios planteados en el apartado anterior se analiza el tiempo que se tarda en realizar la lectura, realizando 30 mediciones de nuevo.

Escenario	TIEMPO DE LECTURA (ms)					
	Baja frecuencia			Alta frecuencia		
	1	2	3	1	2	3
Medida 1	120	103	130	143	144	143
Medida 2	105	108	133	142	143	144
Medida 3	115	111	106	143	143	142
Medida 4	130	126	104	144	144	143
Medida 5	125	125	127	144	144	143
Media	119	114,6	120	143,2	143,6	143

Tabla 21. Resultados de la prueba de tiempo de lectura.

Tiempo de lectura de los sistemas RFID de LF y HF

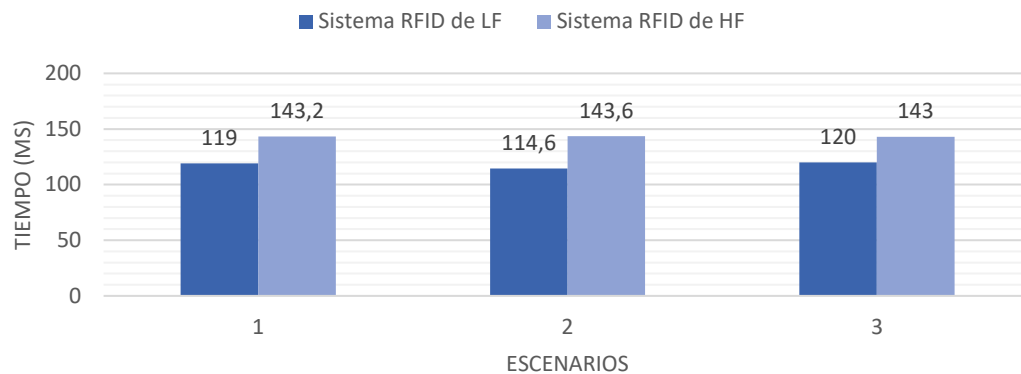


Gráfico 2. Resultados de la prueba de tiempo de lectura.

Al contrario que la distancia de lectura, el tiempo de lectura de etiquetas no se ve afectado por el material del residuo al que esté adherido el *tag* RFID. Es importante saber también, que el tiempo de lectura cambia drásticamente según la cantidad de información que deba ser leída.

7.4.3. Prueba de detección del nivel de llenado

Se realiza esta prueba para comprobar que funcione correctamente con todo tipo de materiales y para ajustar el cálculo del nivel de llenado. Normalmente, el usuario deposita el residuo rápidamente y sin el objetivo de optimizar la distribución del interior del contenedor. Entonces, es habitual que los residuos voluminosos creen espacios vacíos que podrían ser accesibles para residuos de menor tamaño. El sensor, al detectar el objeto más cercano, no tiene en cuenta dichos huecos, por lo que se intentará mejorar el cálculo de nivel de llenado a partir de los resultados de la prueba de detección.

Se realiza la prueba de medición de nivel de llenado con los siguientes residuos:

1. Envase metálico.
2. Envase de cristal translúcido.
3. Envase de plástico transparente.
4. Esponja.
5. Trozo de cartón.



Imagen 33. Residuos utilizados para la prueba de detección del nivel de llenado.

Se introduce cada residuo en el contenedor 5 veces (25 mediciones en total) y se observa si el residuo ha sido detectado. Se marca con una “D” si el producto ha sido detectado y “ND” en caso contrario.

Residuo	DETECCIÓN DEL PRODUCTO				
	1	2	3	4	5
Medida 1	D	D	D	ND	D
Medida 2	D	D	D	D	D
Medida 3	D	D	D	ND	D
Medida 4	D	D	D	D	D
Medida 5	D	D	D	D	D
Resultado más frecuente	D	D	D	D	D

Tabla 22. Resultados de la prueba de detección del nivel de llenado.

Se puede observar que el sensor de ultrasonidos detecta correctamente la gran mayoría de materiales. En dos ocasiones el sensor de ultrasónico no ha sido capaz de detectar la esponja, aunque en la mayoría de mediciones sí ha detectado correctamente el residuo. Ello puede ser causado por el hecho de que la esponja ha absorbido la onda y ésta no ha sido capaz de llegar de vuelta al sensor ultrasónico.

8. Valoración económica

En este capítulo se determinará el coste total de la ejecución del proyecto y la inversión realizada en el prototipo.

8.1. Presupuesto del prototipo

En los cálculos del coste del prototipo no se incluyen las horas de trabajo invertidas. Primero se contabilizará el coste del set general del dispositivo para, posteriormente, cuantificar el coste del contenedor con cada uno de los sistemas RFID.

8.1.1. Set general

DESCRIPCIÓN	CANTIDAD	PRECIO	SUBTOTAL
Placa de desarrollo, Arduino UNO	1,00 u	20,00 €/u	20,00 €
Sensor ultrasónico, HC-SR05	1,00 u	6,35 €/u	6,35 €
Pantalla LCD 16x2	1,00 u	13,80 €/u	13,80 €
Módulo WiFi, ESP-01	1,00 u	2,70 €/u	2,70 €
Cable M/H 40cm	25,00 u	0,06 €/u	1,5 €
Cable M/M 15cm	2,00 u	0,19 €/u	0,38 €
Cable USB (A/B) 180cm	1,00 u	3,70 €/u	3,70 €
Placa de prototipo 16x5	1,00 u	2,80 €/u	2,80 €
Contenedor	1,00 u	3,50 €/u	3,50 €
Suma de la partida			54,73 €
21% I.V.A.			11,49 €
TOTAL PARTIDA			66,22 €

8.1.2. Contenedor con sistema RFID de LF

DESCRIPCIÓN	CANTIDAD	PRECIO	SUBTOTAL
Set general	1,00 u	54,73 €/u	54,73 €
Módulo lector 125kHz, RDM6300	1,00 u	4,21 €/u	4,21 €
Tarjeta RFID 125kHz	3,00 u	1,80 €/u	5,40 €
Suma de la partida			64,34 €
21% I.V.A.			13,51 €
TOTAL PARTIDA			77,85 €

8.1.3. Contenedor con sistema RFID de HF

DESCRIPCIÓN	CANTIDAD	PRECIO	SUBTOTAL
Set general	1,00 u	54,73 €/u	54,73 €
Módulo NFC/RFID <i>elecFreaks</i>	1,00 u	22,02 €/u	22,02 €
Tarjeta RFID 13,56MHz	3,00 u	1,60 €/u	4,80 €
Suma de la partida			81,55 €
21% I.V.A.			17,23 €
TOTAL PARTIDA			98,68 €

8.2. Presupuesto del proyecto

El presupuesto del proyecto se ha calculado sin tener en cuenta el I.V.A.

CAPÍTULO	DESCRIPCIÓN	CANTIDAD	PRECIO	SUBTOTAL
Capítulo 3	Introducción			
	Ingeniero Industrial	10,00 h	50,00 €/h	500,00 €
Capítulo 4	Estado del Arte			
	Ingeniero Industrial	80,00 h	50,00 €/h	4.000,00 €
Capítulo 5	Desarrollo de la idea			
	Ingeniero Industrial	60,00 h	50,00 €/h	3.000,00 €
Capítulo 6	Implementación del <i>hardware</i>			
	Ingeniero Industrial	50,00 h	50,00 €/h	2.500,00 €
Capítulo 7	Implementación del <i>software</i>			
	Ingeniero Industrial	80,00 h	50,00 €/h	4.000,00 €
Capítulo 8	Valoración económica			
	Ingeniero Industrial	10,00 h	50,00 €/h	500,00 €
Capítulo 9	Impacto medioambiental			
	Ingeniero Industrial	10,00 h	50,00 €/h	500,00 €
Capítulo 10	Planificación del Trabajo			
	Ingeniero Industrial	5,00 h	50,00 €/h	250,00 €
Capítulo 11	Conclusiones y líneas futuras			
	Ingeniero Industrial	15,00 h	50,00 €/h	750,00 €
Suma de la partida				16.000,00 €
TOTAL PARTIDA				16.000,00 €

9. Impacto medioambiental

En caso de querer deshacerse del contenedor doméstico inteligente por avería o cualquier otro motivo, éste sería considerado un residuo de aparatos eléctricos y electrónicos (en adelante, RAEE). Por lo tanto, es interesante conocer el proceso de reciclaje de este tipo de residuos.

En la sociedad actual, los aparatos eléctricos y electrónicos tienen un papel fundamental. Están presentes y participan en nuestro día a día, tanto en el ámbito laboral como en el ámbito social y residencial. El número de RAEE generados sigue incrementándose debido a que los ciclos de innovación y sustitución de aparatos son cada vez más breves. Es por ello que es necesario intensificar las medidas de gestión de RAEE tanto en la Unión Europea como en España. Para regular el problema de la gestión de este tipo de residuos en 2015 se aprobó el Real Decreto 110/2015, sobre residuos de aparatos eléctricos y electrónicos.

La legislación vigente prioriza la prevención de la generación de RAEE y su preparación para ser reutilizados. Además, el Real Decreto 110/2015 determina las responsabilidades del usuario, de la administración y del productor, estableciendo que este último es el responsable fundamental de los RAEE. Entre otras responsabilidades, está obligado a financiar la recogida y gestión de los residuos generados por los aparatos. Con estas obligaciones se pretende incentivar al productor a mejorar el diseño de los productos para facilitar su desmontaje, reciclaje o reparación, evitando así el abuso de la obsolescencia programada.

La administración, a través de la fundación Ecolec, se encarga de todos los aspectos del proceso de recogida, de realizar campañas de sensibilización sobre los RAEE, de asegurar la prestación del servicio de recogida en todo el país, etc. El papel del ciudadano en el reciclaje de RAEE se limita al transporte y deposición de estos en puntos limpios (instalaciones municipales) o tiendas adheridas al programa impulsado por Ecolec.

En el caso del prototipo de este proyecto, al no haber integrado completamente todos los componentes, su desmontaje es muy sencillo y todos los componentes pueden ser fácilmente reutilizables, ya que pueden ser empleados en muchos otros proyectos de Arduino.

10. Planificación del Trabajo

Al iniciar el proyecto se decidió realizar un diagrama de Gantt para establecer la planificación del trabajo y poder monitorizar mejor la evolución de éste. En la imagen 34, en morado se muestra la planificación inicial y en naranja la duración y ejecución real del trabajo. Por ciertas demoras se decidió solicitar una prórroga del trabajo y poder disponer de más tiempo para finalizar el dispositivo.

Desde el inicio, el proyecto se retrasó una semana debido a que aún no estaba totalmente definido el enfoque del trabajo y se consideró realizar otra alternativa. Posteriormente, el proyecto se vio retrasado de nuevo porque los componentes necesarios no fueron entregados en el plazo establecido. Por último, se estimó erróneamente el tiempo necesario para aprender a utilizar la plataforma Arduino y su lenguaje de programación desde cero.



11. Conclusiones y líneas futuras

Gracias a la ejecución del proyecto se han podido aplicar los conocimientos obtenidos a lo largo de la titulación. Los conocimientos básicos sobre informática han sido cruciales para facilitar el aprendizaje del funcionamiento de la plataforma Arduino. Las competencias obtenidas en la carrera, como el aprendizaje autónomo o la resolución de problemas y toma de decisiones, han sido clave para el desarrollo y la evolución del dispositivo. Se ha tenido una primera toma de contacto con el prototipado de circuitos electrónicos mediante Arduino y se han adquirido nociones básicas del lenguaje de programación C++, totalmente nuevo a nivel personal. Ha sido muy provechoso profundizar en la tecnología RFID y tener una primera experiencia a nivel práctico para asimilar mejor la información del estudio teórico previo.

A continuación se valora el cumplimiento de los objetivos planteados inicialmente:

- El objetivo principal del proyecto se ha cumplido, dado que se ha construido un prototipo de un contenedor doméstico inteligente que recomienda al usuario dónde depositar el residuo.
- Se ha llevado a cabo un extenso análisis de las alternativas tecnológicas de la identificación automática por radiofrecuencia y se han constatado las prestaciones y limitaciones, tanto a nivel teórico como práctico, de las tres bandas frecuenciales relacionadas con el dispositivo, a excepción del estudio práctico de la banda de UHF que no se ha podido realizar por ser inviable económicamente.
- Gracias al exhaustivo análisis de las distintas bandas frecuenciales podemos distinguir claramente las diferencias técnicas y operacionales entre sistemas trabajando en distinta banda de frecuencia. Un dispositivo basado en un sistema RFID de LF ha de ser combinado con la creación de una base de datos debido a que las etiquetas de esta banda frecuencial tienen una memoria muy reducida. También hay que tener en cuenta que las etiquetas no cuentan con códigos de seguridad. En otros proyectos podría ser una característica determinante. Es necesario acercar el residuo casi hasta el contacto para realizar la lectura del *tag* y el proceso se ha de realizar para cada objeto individualmente. El *firmware* y *software* tras los lectores de LF son sencillos pero limitados y su rendimiento no es óptimo. La mayoría de sus componentes son muy económicos y pequeños. Finalmente, es la banda frecuencial que recibe menos apoyo de la industria.

Un sistema RFID de HF ofrece mayor estabilidad, seguridad y versatilidad. Aparte de poder combinar el dispositivo con una base de datos, también da la posibilidad de escribir las etiquetas en fábrica durante la incorporación de éstas a los productos y que la información pueda ser extraída cuando el residuo del producto se vaya a depositar en el contenedor. Además del identificador, los *tags* pueden contener una

gran cantidad de información. La seguridad del sistema es alta, dado que los *tags* pueden emplear la misma tecnología que usan las tarjetas de pago *contactless*, la tecnología NFC. Al igual que en la banda de baja frecuencia, se requiere acercarse al residuo casi hasta el contacto. Sin embargo, se pueden acercar varios productos a la vez y, aunque la lectura no es simultánea, se leen todos los *tags* uno tras otro. El *firmware* y *software* tanto del lector como de las etiquetas son potentes y tienen una estructura muy bien definida. Junto con la banda de UHF es la banda frecuencial que recibe más apoyo de la industria y la comunidad *Maker*.

Por último, un sistema RFID de UHF ofrece prácticamente las mismas prestaciones que se acaban de presentar de la banda de HF. Sin embargo, se pueden distinguir algunas diferencias como el precio, la distancia de lectura y la lectura simultánea. La ventaja del sistema RFID UHF es que no obliga al usuario a acercarse al residuo al lector, ya que puede leer a mayor distancia y puede leer varios productos simultáneamente al instante. Desgraciadamente, el precio de un lector UHF es aproximadamente 10 veces superior al coste de un lector de HF y su integración en el sistema es mucho más compleja, requiriendo componentes adicionales para su alimentación y disipación de calor. El consumo de un sistema UHF también sería mayor. Como se ha comentado, con la banda de HF, la banda de UHF recibe gran apoyo de la industria. Sin embargo, aunque despierte gran interés en la comunidad *Maker*, los lectores UHF para hacer prototipos aún necesitan ser optimizados para ser totalmente accesibles por la comunidad.

En la tabla 23 se muestra un resumen comparativo de todos los argumentos expuestos previamente. Por aspectos técnicos, operacionales y económicos, la banda frecuencial de HF es la más adecuada y con mayor potencial para el diseño del prototipo. Por lo tanto, se puede concluir que el segundo objetivo específico se ha cumplido.

- El coste inicial del prototipo con sistema LF y HF es de 77,85€ y 98,68€, respectivamente. Este coste solo incluye el precio de los componentes. Para calcular el precio final sería necesario añadir los costes de desarrollo, montaje, programación, etc. Habitualmente el precio de venta de un producto se determina multiplicando el coste de fabricación de un producto por 7, con el objetivo de cubrir los gastos de las actividades previamente mencionadas. En este caso, el precio de venta del dispositivo con sistema LF o HF sería de 544,95€ y 690,76€, respectivamente. El precio final excede sobradamente del precio de mercado de los contenedores domésticos, ya que el precio máximo de los modelos de lujo ronda los 400€. Por lo tanto, actualmente, la realización de un dispositivo final es viable técnicamente pero no económicamente.

PARÁMETRO	LF	HF	UHF
Distancia de lectura	Cercana (1cm).	Cercana (3cm).	Media (1-5m).
Velocidad de lectura	Rápida.	Rápida.	Rápida.
Seguridad	Inexistente	Alta.	Alta.
Lectura simultánea	No, no hay criterio en el orden de lectura.	No, si hay varios <i>tags</i> primero se lee uno y después otro hasta leer todos los <i>tags</i> .	Sí, protocolo con anticolisión. Se lee una etiqueta tras de otra, pero casi en el mismo instante.
Lecturas múltiples	Sí, los datos son enviados repetidamente hasta agotar la energía disponible.	Sí, hay un cierto tiempo entre lectura y lectura.	Sí, hay un cierto tiempo entre lectura y lectura, priorizando las etiquetas que no han sido leídas.
Memoria de los tags	Típicamente solo contiene un identificador.	Contiene un identificador, códigos de seguridad y espacio para información adicional.	Contiene un identificador, códigos de seguridad y espacio para información adicional.

Tabla 23. Comparación de las prestaciones de las bandas frecuenciales de LF, HF y UHF.

Sería necesario un proyecto de optimización y reducción de costes para determinar el precio mínimo de un dispositivo final con las alternativas tecnológicas disponibles en la actualidad. Se puede concluir que el tercer objetivo específico también se ha cumplido, dado que se ha podido responder sobre la viabilidad de un dispositivo final a partir de la realización del prototipo.

Finalmente, tras finalizar el prototipo se proponen posibles mejoras o líneas futuras que podrían sumarle valor y que no se han podido realizar por estar fuera del alcance del trabajo, ya sea por aspectos técnicos, económicos o de tiempo:

- Modificar el dispositivo para que sea autónomo e independiente del ordenador. Sería muy interesante porque ofrecería una imagen más cercana al funcionamiento real del supuesto dispositivo final. Para dotar al prototipo de dichas cualidades, se requeriría añadir una alimentación externa y modificar el *software* para recibir los comandos a través de la red.
- Establecer el módulo ESP-01 como microcontrolador principal del dispositivo. Como se ha mencionado en 6.1.4, el chip ESP8266 incorporado en el módulo ESP-01 es un microcontrolador muy popular que cuenta con el amplio apoyo de la comunidad *Maker*, la cual ha diseñado un *plugin* particular para este chip, para el IDE de Arduino. El *plugin* permite programar el *firmware* del ESP8266 utilizando un lenguaje de programación prácticamente idéntico al lenguaje de Arduino y volcarlo directamente mediante el IDE. Además, cuenta con varias librerías que facilitan la gestión de la conectividad WiFi y amplían las prestaciones del chip. Realizando esta modificación se podría ahorrar la placa Arduino, reduciendo los costes del dispositivo y su tamaño. Sin embargo, se debería comprobar que el ESP-01 sea capaz de controlar todos los demás periféricos, al estar limitado con 2 pines GPIO.
- Modificar el prototipo para hacerlo compatible con los sistemas domésticos existentes como *Alexa*, *Google Home* y *Apple Homekit*. La posibilidad de que el contenedor se pueda integrar en los sistemas domóticos lo introduce completamente en la Internet de las cosas para poder comunicarse con otros dispositivos. Dotaría de valor adicional al prototipo.
- Implementar el sistema RFID de UHF y darle un nuevo enfoque al contenedor que justifique su diseño con esta banda frecuencial. Por ejemplo, podría ser adecuado para la construcción de un contenedor industrial de gran tamaño para empresas que operan con grandes volúmenes de productos (empresas de *retail*, grandes supermercados, etc.)
- Aprovechar la memoria de las etiquetas para incluir más información del residuo.

Datos como su peso, tipología de los materiales que lo conforman o fechas de caducidad (en el caso de los productos alimentarios) podrían ser muy útiles. La inclusión de mayor número de datos permitiría dar más funcionalidades al dispositivo sin necesidad de incorporar más componentes ni alterar su coste. Un ejemplo sería la monitorización de residuos generados. Además se generaría un gran cantidad de datos que podrían ser beneficiosos para optimizar la gestión de residuos.

- Darle un papel más relevante a la conexión a la red e impulsar la participación del dispositivo en el IoT, creando un sistema de avisos del nivel de llenado al móvil. La conexión a la red permitiría el acceso a una posible base de datos con la información necesaria para el reciclaje de residuos, sistema muy similar al ONS de EPCglobal.

Agradecimientos

Para la realización de este trabajo ha sido esencial la ayuda y el asesoramiento de su director, el profesor José Luís Eguía, a quien agradezco su dedicación y disponibilidad a lo largo de todo el proyecto.

Bibliografía

- Aguirre, J. I. (2007). *EPCglobal: A Universal Standard*. (February), 89.
- Albacete, A. De. (2006). *Ayuntamiento de Albacete Guía didáctica sobre Gestión de los Residuos Urbanos en Albacete 1*. 1-39.
- Arduino Uno Rev3 | Arduino Official Store. (s. f.). Recuperado 22 de abril de 2020, de <https://store.arduino.cc/arduino-uno-rev3>
- Beneficios del reciclaje de envases | Ecoembes. (s. f.). Recuperado 20 de noviembre de 2019, de <https://www.ecoembes.com/es/ciudadanos/ecoembes-y-el-medio-ambiente/beneficios-reciclaje>
- Boards & Modules - Arduino. (s. f.). Recuperado 22 de abril de 2020, de <https://store.arduino.cc/arduino-genuino/boards-modules>
- Ciáurriz, D. (2010). Uso y Aplicaciones de las Etiquetas RFID. *Universidad Publica de Navarra*. Recuperado de <http://academica-e.unavarra.es/bitstream/handle/2454/2483/577350.pdf?sequence=1>
- Clau, A. (2006). *envasos alimentaris*. 10(diciembre), 42-50.
- Cruz, S. E., & Ojeda, S. (2013). Gestión sostenible de los residuos sólidos urbanos. *Revista Internacional de Contaminacion Ambiental*, 29(SUPPL. 3), 7-8. <https://doi.org/10.1016/j.habitatint.2014.06.019>
- Curcio, R., Blanco, P., Gil, R., & Rosa, E. (2015). El reciclaje como alternativa de manejo de los residuos sólidos en el sector minas de Baruta, Estado Miranda, Venezuela. *Revista de Investigación*, 39(86), 157-170.
- Elty Eletronika. (2019). *RDM630 Specification*. Recuperado de <https://elty.pl/upload/download/RFID/RDM630-Spec.pdf>
- EM4100 protocol description. (s. f.). Recuperado 22 de abril de 2020, de http://www.priority1design.com.au/em4100_protocol.html
- Evolución de la población mundial hasta 2030 | Statista. (s. f.). Recuperado 20 de noviembre de 2019, de <https://es.statista.com/estadisticas/635122/evolucion-de-la-poblacion-mundial-hasta-2030/>
- GitHub - itead/ITEADLIB_Arduino_WeeESP8266. (s. f.). Recuperado 22 de abril de 2020, de https://github.com/itead/ITEADLIB_Arduino_WeeESP8266
- Glouche, Y., Couderc, P., Glouche, Y., Couderc, P., Waste, A. S., Leister, S., & Couderc, P. (2014). *A Smart Waste Management with Self-Describing objects To cite this version : HAL Id : hal-00924270 A Smart Waste Management with Self-Describing objects*.
- González, A. P. (2016). *RFID*.

López, T., Sarrate, S., Suances, O., Teijeira, C., & Trigo, M. (2010). Identificación por radiofrecuencia (rfid) en alimentos. *Seguridad alimentaria*, 72-84. Recuperado de <http://www.colvema.org/pdf/7284etiquetado.pdf>

MÉS I AMB MILLOR QUALITAT Recollir més i amb millor qualitat. (s. f.).

Mohd Yusof, N., Jidin, A. Z., & Rahim, M. I. (2017). Smart Garbage Monitoring System for Waste Management. *MATEC Web of Conferences*, 97(January). <https://doi.org/10.1051/mateconf/20179701098>

PREMET25 – Programa metropolità de prevenció i gestió de recursos i residus municipals / 2019-2025. (s. f.). Recuperado 20 de noviembre de 2019, de <https://www.premet25.cat/>

Rodríguez, C. P., Del, T., & Alimentos, E. D. E. (2015). *Trazabilidad del estado de alimentos utilizando tags uhf.*

Simultaneous RFID Tag Reader Hookup Guide. (s. f.). Recuperado 22 de abril de 2020, de <https://learn.sparkfun.com/tutorials/simultaneous-rfid-tag-reader-hookup-guide/introduction>

TUTORIAL-TAGS-ISO-14443A-PUNTO-FLOTANTE-S.A. (s. f.). Recuperado 22 de abril de 2020, de <http://www.puntoflotante.net/TARJETAS-TAGS-RFID-ISO-14443A-MFRC522.htm>

Anexos

Anexo I: Encuesta.....	90
Anexo II: Código <i>LecturaLF</i>	91
Anexo III: Código <i>EscrituraHF</i>	95
Anexo IV: Código <i>LecturaHF</i>	98

Anexo I: Encuesta definición del usuario

Las respuestas de la encuesta se adjuntan en archivo Excel: *Anexo I – Respuestas encuesta definición del usuario*. Se puede acceder a la encuesta mediante el link: [https://docs.google.com/forms/d/e/1FAIpQLScXc9489GSgl8X-XeLDYO-WQuhRrVY8hqu_oR5BHmPWPGp-4Q/viewform?usp=sf link](https://docs.google.com/forms/d/e/1FAIpQLScXc9489GSgl8X-XeLDYO-WQuhRrVY8hqu_oR5BHmPWPGp-4Q/viewform?usp=sf_link)

Anexo II: Código *LecturaLF*

El código completo también se incluye, dentro de la carpeta Anexo II, en archivo Arduino: *LecturaLF*

```
// Se incluyen las librerías necesarias para el funcionamiento del dispositivo.
#include <BitbloqLiquidCrystal.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include "ESP8266.h"

// Se crean los objetos LCD, módulo WiFi (ESP8266) y Lector LF (RDM6300); y se
// declaran todas las variables globales.

String estado_sistema;

//-----Pantalla LCD-----

LiquidCrystal LCD(0);

//-----Sensor ultrasonido-----

int pin_TRIG = 9; // Pin para enviar pulsos.
int pin_ECHO = 8; // Pin para recibir la señal de rebote.

unsigned long tiempoSenal;
int distanciaCM;

//-----Módulo WiFi (ESP8266)-----

SoftwareSerial ESP(3, 4); // RX | TX
ESP8266 wifi(ESP);

String option;
String wifi_name;
String wifi_password;

//-----Sensor RFID-----

SoftwareSerial RFID(2, 3);
int card[14] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0};

//-----Iniciamos la base de datos----- // Cada tag ha sido asociado a
// un tipo de residuo distinto.

int tag1[14] = {2,48,48,48,48,48,48,52,51,48,53,52,54,3}; // Botella de
plástico.
int tag2[14] = {2,48,49,49,48,48,53,67,68,54,56,66,49,3}; // Caja de cartón.
int tag3[14] = {2,48,49,49,48,48,53,67,68,50,54,70,70,3}; // Botella de vidrio.

// Función que se ejecuta al iniciar el dispositivo.
void setup(){
  Serial.begin(9600); // Iniciar comunicación entre el PC y Arduino.
  // Configuración inicial de la pantalla LCD.
  LCD.begin(16, 2);
  LCD.setBacklight(HIGH);
  LCD.clear();
  printLCD("Modo Lectura LF",0);
  // Configuración inicial del sensor ultrasónico.
  pinMode(pin_TRIG, OUTPUT);
  pinMode(pin_ECHO, INPUT);
  digitalWrite(pin_TRIG, LOW);
```

```

delayMicroseconds(2);

ESP.begin(9600); // Iniciar comunicació entre el ESP-01 y Arduino.
RFID.begin(9600); // Iniciar comunicació entre el lector HF y Arduino.
}

void loop() {
  if (Serial.available() > 0) {
    estado_sistema = Serial.readString();
    if (estado_sistema.substring(0, estado_sistema.length() - 2) == "Configurar
Wifi") {
      LCD.clear();
      printLCD("Configurando", 0);
      printLCD("Wifi", 1);
      ESP.listen();
      conectWifi(); // Se ejecuta la función para conectarse a una red WiFi.
    }
  }
  ultrasonicSensor(); // Se ejecuta la función del sensor ultrasónico.
  static unsigned char i=0;
  int previoustag[14] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // Utilizado para
comparar otras etiquetas con la etiqueta previa.
  RFID.listen();
  while(RFID.available()) {
    char c = RFID.read();
    if(c==2) { // Si se lee un byte de valor se inicia la lectura del
identificador (según especificación del lector RDM6300).
      card[0]=c;
      i=1;
    }
    else {
      card[i]=c;
      i++;
      if(c==3) { // Si el byte es de valor 3 se ha completado la lectura de la
etiqueta (según especificación del lector RDM6300).
        // Se compara la etiqueta leída con la etiqueta leída previamente. Si
no es la misma se compara con la base de datos.
        if (compareTag(previoustag, card) == false) {
          unsigned char p;
          for(p=0; p<14; p++) {
            previoustag[p]=card[p];
          }
          checkDataBase();
          delay(5000);
          LCD.clear();
          printLCD("Modo lectura LF", 0);
          i=0;
        }
        else {
          i=0;
        }
      }
    }
  }
}

// Función para comparar la etiqueta leída con la base de datos.
void checkDataBase() {
  if (compareTag(card, tag1) == true) {
    Serial.println("Botella de plástico");
    Serial.println("Depositar en el contenedor AMARILLO");
    LCD.clear();
    printLCD("AMARILLO", 0);
  }
}

```

```

    else if (compareTag(card, tag2) == true){
        Serial.println("Caja de cartón");
        Serial.println("Depositar en el contenedor AZUL");
        LCD.clear();
        printLCD("AZUL",0);
    }
    else if (compareTag(card, tag3) == true){
        Serial.println("Botella de vidrio");
        Serial.println("Depositar en el contenedor VERDE");
        LCD.clear();
        printLCD("VERDE",0);
    }
    else{
        Serial.println("Lectura errónea");
        LCD.clear();
        printLCD("Lectura",0);
        printLCD("errónea",1);
    }
}

// Función para comparar el ID de dos etiquetas.
boolean compareTag(int aa[14], int bb[14]){
    boolean ff = false;
    int fg = 0;
    unsigned char p;
    for (int cc = 0 ; cc < 14 ; cc++){
        if (aa[cc] == bb[cc]){
            fg++;
        }
    }
    if (fg == 14){
        ff = true;
    }
    return ff;
}

// Función que muestra un string en una de las líneas de la pantalla LCD.
void printLCD(String txt, const int linea){
    int i=0;
    int longtxt=txt.length();
    while( i < longtxt) { //Mientras no haya timeout
        LCD.setCursor(i, linea);
        LCD.print(txt.charAt((i)));
        i++;
    }
}

// Función para calcular el nivel de llenado.
void ultrasonicSensor() {
    digitalWrite(pin_TRIG, HIGH); // Envía el pulso para iniciar el sensor.
    delayMicroseconds(10);
    digitalWrite(pin_TRIG, LOW); // Se detiene el envío del pulso.
    tiempoSenal = pulseIn(pin_ECHO, HIGH); // Tiempo que dura el pin ECHO en
    HIGH.
    distanciaCM = (0.0343 * tiempoSenal) / 2; // Convierte el tiempo en distancia
    (cm)
    float porcentaje = (32-distanciaCM)*100/35; // Convierte la distancia en
    porcentaje de la tapa hasta la base contenedor-->32cm.
    int parte_entera= int(porcentaje); // Selecciona la parte entera del
    porcentaje.
    if (parte_entera<0){ // La distancia es superior a la distancia hasta la base
    o la señal no es recibida de vuelta.
        printLCD("Cerrar tapa      ",1);
        delay(500);

```

```

    }
    else {
        String string_lcd = "Nvl. lleno: "+String(parte_entera)+"% ";
        printLCD(string_lcd,1); //Muestra en la pantalla LCD el porcentaje.
        delay(500);
    }
}

// Función para conectar a una red WiFi.
void conectWifi(){
    Serial.println("¿Necesitas conectar el contenedor a Internet? (Responda con Yes/No)");
    while (Serial.available()==0) { // Esperamos a que el usuario introduzca la respuesta en el monitor serie.
    }
    option=Serial.readString(); // Leemos la respuesta.
    if(option.substring(0,option.length()-2)=="Yes") {
        Serial.println("Escriba el nombre de la red WiFi a la que quiere conectarse");
        while (Serial.available()==0){ // Esperamos a que el usuario introduzca la respuesta en el monitor serie.
        }
        wifi_name = Serial.readString();
        Serial.println("Escriba la contraseña de la red WiFi a la que quiere conectarse");
        while (Serial.available()==0){ // Esperamos a que el usuario introduzca la respuesta en el monitor serie.
        }
        wifi_password = Serial.readString();
        wifi_name = wifi_name.substring(0,wifi_name.length()-2);
        wifi_password = wifi_password.substring(0,wifi_password.length()-2);
        if (wifi.joinAP(wifi_name, wifi_password)) { // El ESP-01 se ha conectado a la red WiFi introducida.
            Serial.print("Join AP success\r\n");
            Serial.print("IP:");
            Serial.println(wifi.getLocalIP().c_str());
            Serial.println("Conexión lista");
            LCD.clear();
            printLCD("Modo Lectura HF",0);
        }
        else { // El ESP-01 no ha podido conectarse a la red WiFi introducida.
            Serial.print("Join AP failure\r\n");
            Serial.println("No se ha podido conectar a la red, volviendo al modo lectura HF");
            LCD.clear();
            printLCD("Modo Lectura HF",0);
        }
    }
    else if(option.substring(0,option.length()-2)=="No") { // Se comprueba la conexión a la red.
        Serial.print("IP:");
        Serial.println(wifi.getLocalIP().c_str());
        Serial.println("Conexión lista");
        LCD.clear();
        printLCD("Modo Lectura HF",0);
    }
    else{ // No se ha respondido con Yes o No.
        Serial.println("Volviendo al modo lectura HF");
        LCD.clear();
        printLCD("Modo Lectura HF",0);
    }
}
}

```

Anexo III: Código *EscrituraHF*

El código completo también se incluye, dentro de la carpeta Anexo III, en archivo Arduino: *EscrituraHF*

```
// Se incluye la librería del lector HF (PN532).
#include <PN532.h>

//-----Lector HF (PN532)-----
// Macros.
#define SCK          13
#define MOSI         11
#define SS           10
#define MISO         12

PN532 NFC(SCK,MISO,MOSI,SS);

// Declaracion de variables globales.
String productFractionColor; // Color de la fracción a la que debe ir el
producto.
int addedSpaces; // Espacios que hay que añadir al color de la fracción a la
que debe ir el producto.
String block62; // Bloque en el que se almacena el color de la fracción a la
que debe ir el producto.
byte blockcontent[16]; // Informacion a guardar en un bloque.

// Función que se ejecuta al iniciar el dispositivo.
void setup () {
    Serial.begin(9600);
    // Iniciar comunicación entre el lector HF y Arduino.
    if(!Serial) {
        delay(100);
    }
    NFC.begin();
    uint32_t versiondata = NFC.getFirmwareVersion();
    if( ! versiondata ) {
        while (1) {
            delay (100);
        }
    }
    // Configuramos el lector HF para la lectura y escritura.
    NFC.SAMConfig();
    Serial.println("Listo");
}

// Función que se ejecuta reiteradamente.
void loop () {
    Serial.println("Introduzca el color de la fracción a la que el producto debe
ir depositado:");
    while (Serial.available()==0) {
    }
    productFractionColor=Serial.readString();
    productFractionColor=productFractionColor.substring(0,productFractionColor.le
ngth()-2);
    // Si el producto tiene menos de 16 caracteres se rellena con espacios en
blanco
    if(productFractionColor.length()<16){
        addedSpaces=16-productFractionColor.length(); // Se calcula cuantos
espacios hay que añadir al nombre de producto
        for (int i=0; i <addedSpaces ; i++){
            productFractionColor=productFractionColor+' ';
        }
    }
}
```

```

    }
    // En caso contrario se corta la cadena, dejando los 16 primeros caracteres.
    else{
        productFractionColor=productFractionColor.substring(0,16);
    }
    // Se guardará el color de la fracción en el bloque 62.
    block62=productFractionColor;
    Serial.println("////////////////////////////////////");
    Serial.println("Se va a escribir la tarjeta con la informacion que se acaba
de introducir");
    Serial.println("Aproxime una tarjeta"); //En espera de que se aproxime una
tarjeta al lector
    // Leemos el número de identificación, mientras no haya ninguna etiqueta
valdrá 0
    uint32_t id = NFC.readPassiveTargetID(PN532_MIFARE_ISO14443A);
    uint8_t key [ ] = { 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF} ; //Clave por defecto

    while (id==0){
        id = NFC.readPassiveTargetID(PN532_MIFARE_ISO14443A);
    }
    if (id){
        StartTime=millis();
        Serial.print("Tarjeta Detectada, ID:");
        Serial.println(id);
        // Escritura de la tarjeta.
        // Guardamos el color de la fracción en blockcontent.
        fillBlock(blockcontent,productFractionColor);
        confirmBlock(62, blockcontent);
        // Escribimos el contenido de blockcontent en el bloque 62 de nuestra
tarjeta
        if (NFC.authenticateBlock(1, id, 62, KEY_A, key)){
            NFC.writeMemoryBlock(1, 62, blockcontent);
        }
    }
}

// Función para añadir ceros a la izquierda.
String leftZeros(String num, int zeros){
    while (num.length()<zeros){
        num='0'+num;
    }
    return num;
}

// Función para añadir espacios a la derecha.
String rightBlank(String str, int blanks){
    while (str.length()<blanks){
        str=str+' ';
    }
    return str;
}

// Función para rellenar un bloque.
void fillBlock(byte blockcontent[], String str){
    for (int i=0; i <16 ; i++){
        blockcontent[i]=str[i];
    }
}

// Función para asegurarse que solo se escriba en bloques de datps y no en
bloques de seguridad.
int confirmBlock(int blockNumber, byte arrayAddress[]){
    int largestModulo4Number=blockNumber/4*4;

```

```
int trailerBlock=largestModulo4Number+3; // Determina el respectivo bloque
del sector trailer del sector.
if (blockNumber > 2 && (blockNumber+1)%4 ==
0){Serial.print(blockNumber);Serial.println(" is a trailer block:");return 2;}
Serial.print(blockNumber);
Serial.println(" is a data block:");
}

//El código se ha adaptado de este tutorial
http://makecourse.weebly.com/week10segment1.html
```

Anexo IV: Código *LecturaHF*

El código completo también se incluye, dentro de la carpeta Anexo IV, en archivo Arduino: *LecturaHF*

```
// Se incluyen las librerías necesarias para el funcionamiento del dispositivo.
#include <BitbloqLiquidCrystal.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include "ESP8266.h"
#include <PN532.h>

// Se crean los objetos LCD, módulo WiFi (ESP8266) y Lector HF (PN532); y se
// declaran todas las variables globales.

String comando;

//-----Pantalla LCD-----

LiquidCrystal LCD(0);

//-----Sensor ultrasonido-----

int pin_TRIG = 9; // Pin para enviar pulsos.
int pin_ECHO = 8; // Pin para recibir la señal de rebote.

unsigned long tiempoSenal;
int distanciaCM;

//-----Módulo WiFi (ESP8266)-----

SoftwareSerial ESP(3, 4); // RX | TX
ESP8266 wifi(ESP);

String option;
String wifi_name;
String wifi_password;

//-----Lector HF (PN532)-----
// Macros
#define SCK          13
#define MOSI         11
#define SS           10
#define MISO         12

PN532 NFC(SCK,MISO,MOSI,SS);

int addedSpaces; // Espacios que hay que añadir al color de la fracción a la que
// debe ir el producto.
String block62; // Bloque en el que se almacena el color de la fracción a la que
// debe ir el producto.
byte blockcontent[16]; // Información a guardar en un bloque.

// Función que se ejecuta al iniciar el dispositivo.
void setup() {
    Serial.begin(9600); // Iniciar comunicación entre el PC y Arduino.
    // Configuración inicial de la pantalla LCD.
    LCD.begin(16, 2);
    LCD.setBacklight(HIGH);
    LCD.clear();
    printLCD("Modo Lectura HF",0);
    // Configuración inicial del sensor ultrasónico.
```

```

pinMode(pin_TRIG, OUTPUT);
pinMode(pin_ECHO, INPUT);
digitalWrite(pin_TRIG, LOW);
delayMicroseconds(2);

ESP.begin(9600); // Iniciar comunicación entre el ESP-01 y Arduino.
// Iniciar comunicación entre el lector HF y Arduino.
if(!Serial){
    delay(100);
}
NFC.begin();
uint32_t versiondata = NFC.getFirmwareVersion();
if( ! versiondata ) {
    while (1) {
        delay (100);
    }
}
// Configuramos el lector HF para la lectura y escritura.
NFC.SAMConfig();
Serial.println("Listo, esperando a leer productos");
}

// Función que se ejecuta reiteradamente.
void loop() {
    // Se definen las variables que guardan el ID y la Key B de los tags MIFARE.
    uint32_t id = NFC.readPassiveTargetID(PN532_MIFARE_ISO14443A);
    uint8_t key [ ] = { 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}; //Clave por defecto

    while (id==0){ // Mientras no haya etiquetas dentro del rango del lector.
        ultrasonicSensor(); // Se ejecuta la función del sensor ultrasónico.
        id = NFC.readPassiveTargetID(PN532_MIFARE_ISO14443A);
        if (Serial.available()>0){
            comando = Serial.readString();
            if (comando.substring(0,comando.length()-2)=="Configurar Wifi"){
                LCD.clear();
                printLCD("Configurando",0);
                printLCD("Wifi",1);
                conectWifi(); // Se ejecuta la función para conectarse a una red WiFi.
            }
        }
    }
    if (id){ // Se ejecuta cuando el lector ha detectado una o más etiquetas.
        Serial.print ("\n");
        Serial.print("Tarjeta Detectada, ID:");
        Serial.println(id);
        String productFractionColor; // Variable que almacena el color de la
        fracción donde depositar el residuo.
        // Se extraen los datos contenidos en el bloque 62 de la tarjeta.
        if (NFC.authenticateBlock(1, id, 62, KEY_A, key)){
            uint8_t block[16];
            if (NFC.readMemoryBlock (1, 62, blockcontent) ) {
                fillString(blockcontent, block62);
                for( int i = 0; i<16 ; i++){
                    Serial.write(blockcontent[i]);
                    productFractionColor = productFractionColor+String(char
                    (blockcontent[i]));
                }
                Serial.print ("\n");
            }
        }
        LCD.clear();
        printLCD(productFractionColor,0);
        delay(2000);
        printLCD("Modo Lectura HF",0);
    }
}

```

```

    }
}

// Función que muestra un string en una de las líneas de la pantalla LCD.
void printLCD(String txt, const int linea){
    int i=0;
    int longtxt=txt.length();
    while( i < longtxt) { //Mientras no haya timeout
        LCD.setCursor(i, linea);
        LCD.print(txt.charAt((i)));
        i++;
    }
}

// Función para rellenar un bloque.
void fillString(byte blockcontent[], String str){
    for (int i=0; i <16 ; i++){
        str=str+String(blockcontent[i], HEX);
    }
}

// Función para calcular el nivel de llenado.
void ultrasonicSensor() {
    digitalWrite(pin_TRIG, HIGH); // Envia el pulso para iniciar el sensor.
    delayMicroseconds(10);
    digitalWrite(pin_TRIG, LOW); // Se detiene el envío del pulso.
    tiempoSenal = pulseIn(pin_ECHO, HIGH); // Tiempo que dura el pin ECHO en
HIGH.
    distanciaCM = (0.0343 * tiempoSenal) / 2; // Convierte el tiempo en distancia
(cm)
    float porcentaje = (32-distanciaCM)*100/35; // Convierte la distancia en
porcentaje; distancia de la tapa hasta la base contenedor-->32cm.
    int parte_entera= int(porcentaje); // Selecciona la parte entera del
porcentaje.
    if (parte_entera<0){ // La distancia es superior a la distancia hasta la base
o la señal no es recibida de vuelta.
        printLCD("Cerrar tapa      ",1);
        delay(500);
    }
    else {
        String string_lcd = "Nvl. lleno: "+String(parte_entera)+"% ";
        printLCD(string_lcd,1); //Muestra en la pantalla LCD el porcentaje.
        delay(500);
    }
}

void conectWifi(){
    Serial.println("¿Necesitas conectar el contenedor a Internet? (Responda con
Yes/No)");
    while (Serial.available()==0) { // Esperamos a que el usuario introduzca la
respuesta en el monitor serie.
    }
    option=Serial.readString(); // Leemos la respuesta.
    if(option.substring(0,option.length()-2)=="Yes") {
        Serial.println("Escriba el nombre de la red WiFi a la que quiere
conectarse");
        while (Serial.available()==0){ // Esperamos a que el usuario introduzca la
respuesta en el monitor serie.
        }
        wifi_name = Serial.readString();
        Serial.println("Escriba la contraseña de la red WiFi a la que quiere
conectarse");
        while (Serial.available()==0){ // Esperamos a que el usuario introduzca la
respuesta en el monitor serie.

```

```
    }
    wifi_password = Serial.readString();
    wifi_name = wifi_name.substring(0,wifi_name.length()-2);
    wifi_password = wifi_password.substring(0,wifi_password.length()-2);
    if (wifi.joinAP(wifi_name, wifi_password)) { // El ESP-01 se ha conected a
la red WiFi introducida.
        Serial.print("Join AP success\r\n");
        Serial.print("IP:");
        Serial.println(wifi.getLocalIP().c_str());
        Serial.println("Conexión lista");
        LCD.clear();
        printLCD("Modo Lectura HF",0);
    }
    else { // El ESP-01 no ha podido conectarse a la red WiFi introducida.
        Serial.print("Join AP failure\r\n");
        Serial.println("No se ha podido conectar a la red, volviendo al modo
lectura HF");
        LCD.clear();
        printLCD("Modo Lectura HF",0);
    }
}
else if(option.substring(0,option.length()-2)=="No") { // Se comprueba la
conexión a la red.
    Serial.print("IP:");
    Serial.println(wifi.getLocalIP().c_str());
    Serial.println("Conexión lista");
    LCD.clear();
    printLCD("Modo Lectura HF",0);
}
else{ // No se ha respondido con Yes o No.
    Serial.println("Volviendo al modo lectura HF");
    LCD.clear();
    printLCD("Modo Lectura HF",0);
}
}
```

Anexo V: Diagrama de Gantt

El diagrama de Gantt de la planificación del trabajo se adjunta en archivo Excel: *Anexo V – Planificación del Trabajo de Fin de Grado*.